FACULTAD DE ESTUDIOS ESTADÍSTICOS

MÁSTER EN MINERÍA DE DATOS E INTELIGENCIA DE NEGOCIOS

Curso 2018/2019

Trabajo de Fin de Máster

TITULO: Aplicación de técnicas de minería de datos para la predicción del coste total anual del servicio de salud.

***Alumno*:** Diana María Navarrete Cruz

***Tutores*:** Aida Calviño Martínez
Jaime Mosquera Restrepo

Noviembre de 2019

UNIVERSIDAD COMPLUTENSE
MADRID

Contenido

Índice de Figuras

Índice de Tablas

1. Introducción

El sistema general de seguridad social en salud en Colombia se encuentra bajo la dirección y control del Estado por medio del Ministerio de la salud y protección social y cuenta con dos regímenes de afiliación: contributivo y subsidiado. Este sistema se rige por la Ley 100 de 1993, que establece un esquema de aseguramiento mediante la definición de un Plan de Beneficios en Salud (PBS), cuyos servicios y tecnologías se financian con cargo a la Unidad de Pago por Capitación (UPC), que se reconoce a las Entidades Promotoras de Salud (EPS) por cada persona afiliada. Esta UPC es determinada periódicamente por el Ministerio y asignada a las EPS, teniendo en cuenta características de la población como son la edad y el género.

Las EPS son entidades privadas que actúan como intermediarias y administradoras de los recursos que provee el estado, en forma de prima anual. Con estos recursos, las EPS deben cubrir tanto los costes de prestación del servicio, como los gastos operativos y de administración del riesgo, sin exceder en este último el 10% de los recursos recibidos.

En resumen, como se puede observar en la Figura 1 los afiliados tienen un contrato con la EPS y realizan unos aportes mensuales, estos aportes los recauda la EPS y los envía a la administradora de recursos del sistema general de seguridad social en salud (ADRES). La ADRES dependiendo del número de afiliados realiza un pago de una prima anual. Con estos recursos la EPS debe pagar a las instituciones prestadoras de salud (IPS) por los servicios prestados a los afiliados.

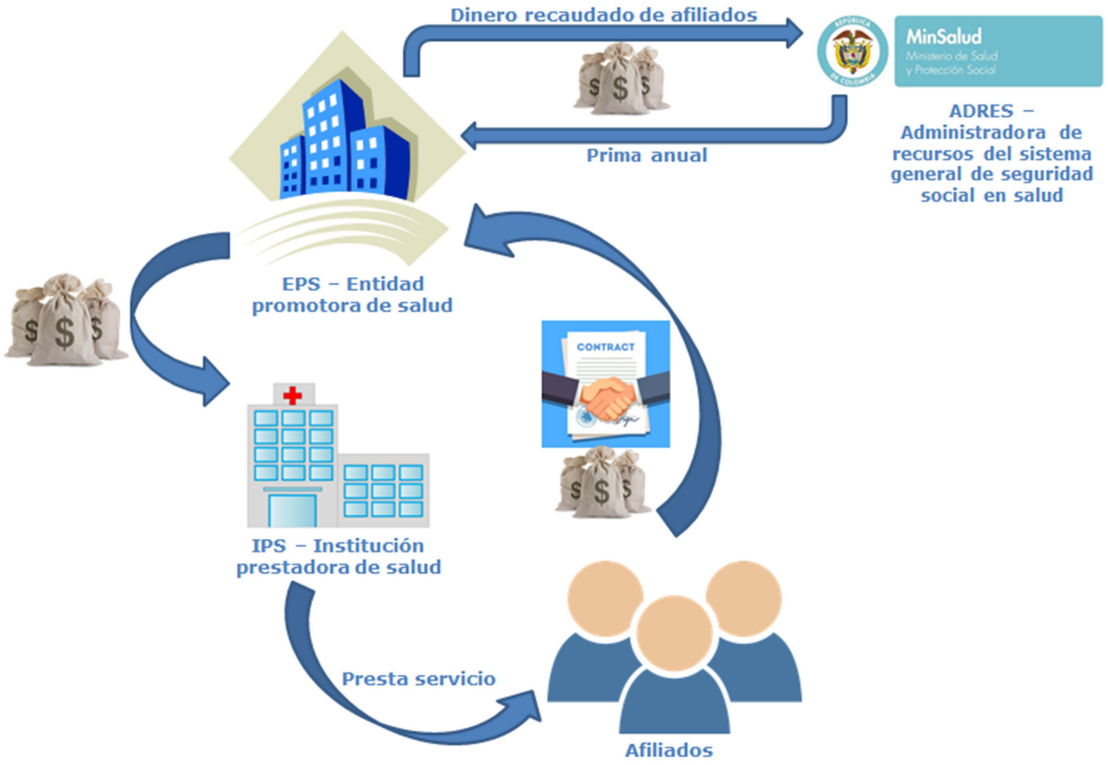


Figura 1. Estructura general del sistema de salud en Colombia.

1.1 Justificación

En el sector de las aseguradoras de los servicios de salud, disponer de información acerca del comportamiento del coste del servicio médico de su población de afiliados y lograr establecer grupos poblacionales que presenten un perfil de consumo relativamente similar, además de identificar posibles factores que generen un incremento o reducción del coste, es de vital importancia porque permite al ente asegurador (EPS) reservar los recursos adecuados para cubrir los costes del servicio y prever posibles déficits presupuestales con respecto al dinero recibido por parte del Gobierno. Para lograr este objetivo, se modelará el coste total anual de prestación del servicio de salud, para el conjunto de afiliados de la EPS. Se dispone de una base de datos con 10.331 registros, obtenidos de forma aleatoria del total de afiliados que supera los dos millones. Se pretende que los resultados de este trabajo sean aplicados en la entidad y puedan ser implementados como herramienta para la toma de decisiones.

1.2 Estado del arte

Para modelar el coste médico, la técnica más utilizada en la literatura son los modelos de regresión. Múltiples autores han realizado estudios en esta materia, realizando pruebas con modelos de una parte y dos partes como Duan et al., (1982), Mullahy, (1998), Diehr et al., (1999), Adams et al., (1993), Lipscomb et al., (1998), Liu et al., (2010) y Mosquera, (2016) entre otros. Los modelos de dos partes consisten en dividir el problema, como su nombre indica, en dos: la primera modelará la probabilidad de ocurrencia del evento $Pr(Y > 0|X)$ y la segunda, predecirá el valor de la variable dependiente, condicionada por la primera parte $(Y|X, Y > 0)$.

Los modelos de dos partes fueron introducidos por Cragg, 1971), y utilizados como uno de los modelos alternativos para predecir la demanda de servicios en salud en Duan et al. (1982). En este libro se plantean cinco modelos alternativos: análisis de varianza (ANOVA), análisis de covarianza (ANCOVA) con costes sin transformar como variable dependiente, un modelo de una parte que utiliza dos parámetros Box-Cox para transformar el coste y los modelos de dos partes utilizando dos ecuaciones separadas para estimar la probabilidad de costes positivos y el nivel de estos costes y un modelo de cuatro partes para predecir los costes cuando involucra hospitalización.

Mullahy (1998) considera transformaciones y modelos de dos partes en econometría de la salud. Recomienda la utilización de una función de enlace logarítmico para la modelación de la segunda parte para así garantizar la positividad de la variable de respuesta.

En Diehr et al. (1999), los autores prueban un modelo de regresión por el método de mínimos cuadrados (OLS) y un modelo de dos partes, utilizando regresión logística para la primera, y regresión lineal para la segunda con distribución log normal y gamma. Las recomendaciones de los autores apuntan a definir claramente los objetivos de la investigación, ya que dependiendo de estos será más útil un modelo que otro. Si el objetivo es entender el sistema, el modelo de dos partes permite detectar los factores que afectan a la propensión del uso del servicio.

Mientras que si el objetivo es predecir los costes futuros, un modelo de regresión bastaría, ya que se obtienen resultados similares al modelo de dos partes pero es más fácil de interpretar.

Adams et al. (1993) y Lipscomb et al. (1998) comparan de nuevo un modelo de regresión por el método de mínimos cuadrados (OLS), dos modelos de dos partes con regresión logística para la primera, uno con regresión de Cox para la primera parte y otro con un modelo paramétrico asumiendo distribución Weibull. Además examinan los efectos de las variables de entrada en las predicciones del coste, para determinar cuáles son más significativas. Concluyen que el modelo con la predicción más precisa es el de dos partes con regresión de Cox.

En Liu et al. (2010), los autores prueban el modelo de dos partes con regresión logística y para la parte continua adicionan a la regresión efectos aleatorios.

También se ha utilizado la simulación en la predicción del coste (Joyanes-Aguilar et al. 2015) donde se utilizó minería de datos para la elaboración de predictores, y se construyó un entorno de simulación para determinar el coste económico de la evolución diagnóstica. En Vargas y Giraldo (2014) utilizaron simulación discreta, evaluando diferentes escenarios de prestación del servicio de salud, para predecir el coste de prestación del servicio anual de una EPS.

En Mosquera (2016), se desarrolla un modelo de dos partes utilizando regresión logística para la primera y regresión para la segunda, con función de enlace logarítmica y distribución gamma. Adicionalmente, se prueba un modelo de máquinas de soporte vectorial y un árbol de regresión, para estimar el coste medio anual del servicio de salud. El autor concluye que tanto el modelo de dos partes como el árbol de regresión, presentan un mejor ajuste desde el punto de vista estadístico y financiero. Finalmente, recomienda el modelo de dos partes por la interpretabilidad de los resultados, ya que permite identificar y cuantificar los efectos sobre la propensión al uso que presentan las diferentes variables consideradas sobre la población.

En el desarrollo del presente trabajo se aplicará por tanto, un modelo de dos partes, probando diferentes técnicas de clasificación y regresión, como son: regresión lineal y logística, redes neuronales, random forest, bagging, incremento gradiente, máquinas de soporte vectorial y ensamblado. Se definirá para cada parte la técnica que mejores resultados presente, para finalmente configurar el modelo de dos partes y obtener las predicciones del coste total anual del servicio de salud de la entidad como el producto de las predicciones de las dos partes.

En cuanto a herramientas computacionales, se cuenta con un ordenador con procesador Intel Core i5 a 1.8GHz 2 procesadores principales y 4 lógicos, memoria RAM de 8 GB y sistema operativo Windows 10. Se utilizarán los programas: SAS Base®, SAS Enterprise Miner ® y R.

2. Objetivos

2.1 Objetivo General

Aplicar técnicas de minería de datos para predecir el coste total anual del servicio de salud en una entidad promotora de salud, incluyendo las posibles variables que afecten la predicción.

2.2 Objetivos específicos

- Definir las variables a utilizar en el modelo.
- Crear modelos de predicción para el coste total anual del servicio de salud por diferentes métodos.
- Comparar los diferentes métodos utilizados y concluir cuál es el mejor, identificando las variables más influyentes.

3. Metodología y marco teórico

3.1 SEMMA

Para desarrollar el presente trabajo se utilizará la metodología SEMMA (Sample - Explore – Modify – Model – Assess) propuesta por SAS, que se describe a continuación:

Muestreo: La primera fase de la metodología consiste en extraer una muestra representativa de la población. El método de muestreo más utilizado es el "muestreo aleatorio simple": este método asigna a cada elemento de la población la misma probabilidad de ser seleccionado. Se cuenta con 10331 observaciones, como se verá más adelante.

Exploración: En esta fase se realiza un análisis exploratorio y descriptivo, utilizando herramientas de visualización de datos y diferentes técnicas estadísticas, para descubrir posibles relaciones entre las variables independientes del modelo y la variable objetivo, también permite detectar posibles errores en los datos que serán corregidos en la siguiente fase.

Modificación: El objetivo de esta fase es corregir los errores detectados en la exploración, así como eliminar variables que no aporten valor al modelo, crear nuevas variables a través de transformaciones de las variables de entrada originales, etc.

Modelización: En esta fase se configuran los diferentes modelos que permitan encontrar la relación entre la variable objetivo o dependiente y las variables explicativas, permitiendo predecir el valor que tomará la variable objetivo. Las técnicas que se utilizarán para el modelado de los datos serán: redes neuronales, regresión lineal y logística, bagging, random forest, incremento gradiente, máquinas de soporte vectorial y ensamblado de modelos.

Evaluación: Por último, se realiza una evaluación y comparación de los modelos para determinar la calidad de las predicciones y definir cuál es el modelo que presenta el error más bajo al predecir, este modelo será el que se implementará en la entidad y será utilizado por el área financiera como apoyo en el proceso de planificación y toma de decisiones. Las principales medidas de evaluación para métodos de predicción, cuando la variable objetivo es continua, son las siguientes:

$$MAEp_j = \frac{\sum_{i=1}^{Nobs}|y_i - \hat{y}_{ij}|}{Nobs} \qquad MSEp_j = \frac{\sum_{i=1}^{Nobs}(y_i - \hat{y}_{ij})^2}{Nobs}$$

$$RMSEp_j = \sqrt{\frac{\sum_{i=1}^{Nobs}(y_i - \hat{y}_{ij})^2}{Nobs}} \qquad NMSEp_j = \frac{\sum_{i=1}^{Nobs}(y_i - \hat{y}_{ij})^2}{\sum_{i=1}^{Nobs}(y_i - \hat{y})^2}$$

Para este trabajo se utilizara el ASE (Average Square Error) o MSE, error cuadrático promedio, que, como su nombre indica, mide el promedio de los errores al cuadrado, siendo el error, la diferencia entre el valor obtenido por la predicción y el valor realmente observado.

Para métodos de clasificación, cuyo objetivo es clasificar las observaciones como "evento" o "no evento", se debe decidir a partir de qué valor de probabilidad o punto de corte se considera una observación como evento. El punto de corte más utilizado es 0.5, esto quiere decir que si una observación tiene una probabilidad igual o superior a 0.5 será clasificada como evento. Para estos métodos, se suele usar como medida de evaluación, la matriz de confusión que se muestra en la Figura 2, que contiene el número de observaciones que fueron bien o mal clasificadas:

Predicción	Observado: Positivos (1)	Observado: Negativos (0)
Positivos (1)	Verdaderos positivos (VP)	Falsos Positivos (FP)
Negativos (0)	Falsos Negativos (FN)	Verdaderos Negativos (VN)

Figura 2. Matriz de confusión.

A partir de esta matriz se puede calcular la tasa de acierto, que es el porcentaje del número total de predicciones que fueron correctas.

Tasa de acierto, es la proporción del número total de predicciones que fueron correctas:

$$Tasa\ de\ acierto = \frac{VN + VP}{VN + FP + FN + VP}$$

Tasa de fallo, es la proporción del número total de predicciones que fueron incorrectas:

$$Tasa\ de\ fallo = \frac{FN + FP}{VN + FP + FN + VP}$$

Sensibilidad o tasa de verdaderos positivos, proporción de casos positivos que fueron correctamente clasificados:

$$Sensibilidad = \frac{VP}{FN + VP}$$

Especificidad o tasa de verdaderos negativos, proporción de casos negativos que fueron correctamente clasificados:

$$Especificidad = \frac{VN}{VN + FP}$$

Valor predictivo positivo (VPP), proporción de casos predichos positivos que fueron correctos, esta medida evalúa que tan fiable es la predicción de los valores positivos del modelo:

$$VPP = \frac{VP}{FP + VP}$$

Valor predictivo negativo (VPN), proporción de casos predichos negativos que fueron correctos, esta medida evalúa que tan fiable es la predicción de los valores negativos del modelo:

$$VPN = \frac{VN}{VN + FN}$$

Otra de las medidas de evaluación más utilizadas en modelos de clasificación es la curva ROC. Esta curva muestra gráficamente la sensibilidad (tasa de verdaderos positivos) vs, 1-especificidad (tasa de falsos positivos), para todos los posibles puntos de corte. El modelo perfecto es aquel que asigna probabilidad cero a los no eventos y uno a los eventos. Por lo tanto, la especificidad y sensibilidad serían igual a uno.

El área bajo la curva ROC (llamada AUC) proporciona una medida de la capacidad predictiva de un clasificador. Se puede interpretar como la probabilidad de que un clasificador ordenará o puntuará una instancia positiva elegida aleatoriamente más alta que una negativa. Esta área posee un valor comprendido entre 0.5 y 1, donde 1 representa el valor perfecto y 0.5, un modelo sin capacidad de clasificar. Por esto, siempre se elige el modelo que presente una mayor área bajo la curva. En la Figura 3 se muestran algunos ejemplos.

Como se pretende que este modelo sea utilizado por el área financiera, resulta interesante incluir un indicador de ajuste financiero global, definido en Mosquera (2016), como la diferencia entre el coste total observado y el coste total predicho por el modelo, en términos relativos:

$$IF_p = \frac{\sum_{i=0}^{Nobs}(\hat{y}_{ij} - y_i)}{\sum_{i=1}^{Nobs} y_i}$$

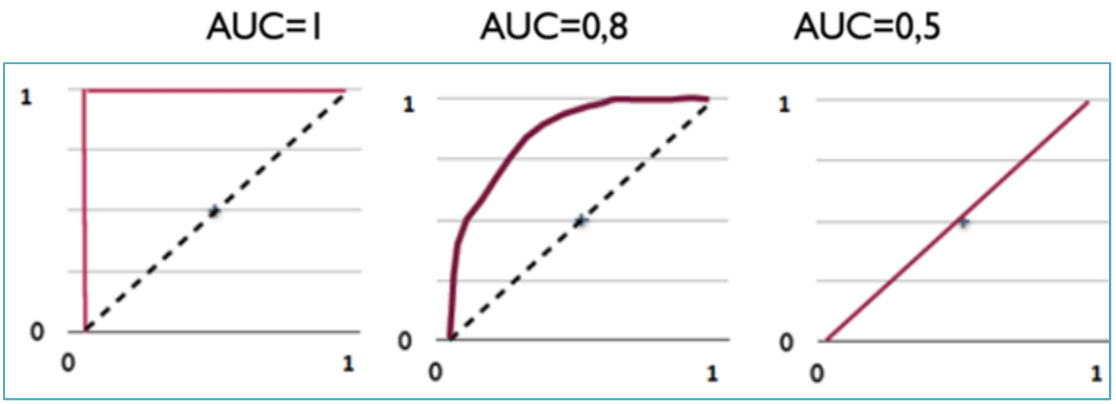


Figura 3. Ejemplos de gráficos ROC. Fuente: Friedman et al. (2009)

A continuación se describen las técnicas de minería de datos que se van a utilizar en el presente trabajo.

3.2 Validación cruzada repetida

La validación cruzada repetida sirve para evaluar los resultados de las técnicas que se van a aplicar, su objetivo es prevenir el sobreajuste de los datos de entrenamiento. Consiste en dividir los datos aleatoriamente en k grupos, dejando un conjunto i que no se utilizará en la construcción del modelo y construyendo el modelo con el resto de grupos (k-i). Este proceso se repite n veces y estimando el error con el conjunto i. En este trabajo se utilizarán 4 grupos.

3.3 Regresión lineal

La regresión lineal tiene por objetivo predecir la variable dependiente (y) a partir de un conjunto de m variables independientes o explicativas x, a través de la ecuación:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_m x_m + \varepsilon$$

donde $\boldsymbol{\varepsilon}$ representa el error o la parte de la variable dependiente no explicada, $\boldsymbol{y}$ es la variable dependiente continua, $\boldsymbol{\beta_0}$ representa el valor que toma la variable dependiente cuando todas las variables independientes toman el valor 0 y los parámetros $\boldsymbol{\beta_i}$ representan cuanto aumenta o disminuye la variable dependiente por cada incremento unitario de la variable i-ésima.

Conociendo los valores de las variables independientes x, se puede predecir el valor de la variable objetivo o dependiente $\boldsymbol{y}$, estimando los valores de los parámetros $\boldsymbol{\beta_i}$ que minimicen el error cometido por el modelo.

3.4 Regresión Logística

Similar a los modelos de regresión lineal, la regresión logística tiene por objetivo predecir la variable dependiente (y) a partir de un conjunto de variables independientes o explicativas (x), con la particularidad de que la variable respuesta o dependiente admite varias categorías de respuesta (politómica). Este tipo de regresión es muy útil en el caso de existir dos posibles respuestas, es decir, cuando la variable dependiente es dicotómica.

El procedimiento que aplica esta técnica es el siguiente: se define una función de enlace que relaciona la variable (y) con las (x), esta puede ser la función logit para el caso de variable objetivo dicotómica, o logit generalizada si la variable dependiente tiene más de 2 categorías. El algoritmo optimiza los parámetros y se obtienen predicciones en forma de probabilidad de pertenecer a cada clase. Es necesario determinar cuál es el criterio o punto de corte (threshold) para asignar cada observación a una clase. Por defecto, el punto de corte es 0.5 (a partir de una probabilidad predicha de pertenecer a la clase A superior a 0.5 se asigna la observación a la clase A).

El modelo de regresión logística presenta la siguiente relación, donde la probabilidad de evento (o que la variable dependiente tome valor uno) es:

$$p_1 = P(Y = 1 \,|x_1, x_2, \dots, x_m) = \frac{1}{1 + \, e^{-(\alpha_0 + \alpha_1 x_1 + \cdots + \alpha_m x_m)}}$$

Por lo cual, dado que la probabilidad de no evento es uno menos la probabilidad de evento, $p_0 = 1 - \, p_1$ se tiene entonces que:

$$P(Y = 0 \,|x_1, x_2, \dots, x_m) = \frac{e^{-(\alpha_0 + \alpha_1 x_1 + \cdots + \alpha_m x_m)}}{1 + e^{-(\alpha_0 + \alpha_1 x_1 + \cdots + \alpha_m x_m)}}$$, Por lo que,

$$\log\left(\frac{p_1}{1 - \, p_1}\right) = \alpha_0 + \alpha_1 x_1 + \cdots + \alpha_m x_m$$

El segundo término recibe el nombre de logit, y es el logaritmo de la razón de probabilidades u odds ratio. Se puede comprobar que utilizando esta función de enlace, las probabilidades resultantes están restringidas al intervalo (0,1).

Nótese que si un determinado parámetro es positivo (o negativo), aumentar una unidad la variable correspondiente se traduce en un aumento (o disminución) de p_1 y del logit (en este último en α_i unidades). Por lo tanto, la exponencial del parámetro representa el odds-ratio asociado a la variable y, que permite aproximar cuánto más probable (o improbable) es que se dé el evento entre los individuos con $x = 1$ frente a los individuos con $x = 0$.

3.5 Modelo de dos partes

Como se mencionó en el estado del arte, este tipo de modelos ha sido ampliamente utilizado en el área de la econometría de la salud, ya que la mayoría de las variables dependientes (y), cumplen con dos propiedades estadísticas fundamentales: son variables de respuesta estrictamente positivas y cuentan con una alta proporción de valores cero. Como alternativa para modelar estas variables Duan et al. (1982) plantea el modelo de dos partes (TPM), en este modelo se define que la primera parte del modelo está relacionada con la probabilidad de que la variable respuesta sea mayor que cero $Pr(Y > 0|X)$ y es gobernada por un modelo de probabilidad binario como el logit o probit. La segunda parte, condicionada por la primera $(Y|X, Y > 0)$, corresponde a una variable continua, generalmente de distribución asimétrica positiva. El valor esperado de la variable de respuesta dados los valores de x, se puede calcular como:

$$E(Y|X) = P\ (Y > 0|X) * E(Y|X, Y > 0)$$

Donde la expresión $P\ (Y > 0|X)$, corresponde a la probabilidad de ocurrencia del evento. Para el caso de la regresión logística correspondería con:

$$P\ (Y > 0|X) = \frac{1}{1+ e^{-(\alpha_0+\alpha_1 x_1+\cdots+\alpha_m x_m)}} = \frac{1}{1+ e^{-X\alpha}} = \frac{1}{1+\frac{1}{e^{X\alpha}}} = \frac{e^{X\alpha}}{1+e^{X\alpha}},$$

$$\text{donde } X = (x_1 + x_2 + \cdots + x_m).$$

La segunda parte de la ecuación $E(Y|X, Y > 0)$, dado que se requiere que sea positiva la variable de respuesta, si se modela con regresión se deberá utilizar alguna función de enlace que garantice su positividad, por ejemplo el enlace logarítmico nombrado en los artículos de Mullahy (1998) y Manning y Mullahy (2001).

De acuerdo con lo anterior, las dos partes del modelo pueden ser estimadas individualmente. Para esto, se preparan dos bases de datos, una que contenga la variable dependiente expresada en ceros y uno (binaria) si el coste es cero y uno si tiene algún coste, y la segunda, contendrá exclusivamente los registros en los que la variable dependiente sea diferente de cero.

3.6 Métodos basados en árboles

Los métodos como incremento gradiente, bagging y random forest se basan en promediar la salida de varios árboles de clasificación o de regresión.

Los árboles son métodos computacionales intensivos que segmentan la información aplicando de forma jerárquica y secuencial una serie de reglas. Cada segmento se denomina nodo y contiene un subconjunto de las observaciones. Este método trata de conseguir homogeneidad entre los valores de la variable respuesta en el mismo nodo y heterogéneos entre los demás nodos. El procedimiento inicia en el nodo raíz que contiene la totalidad de las observaciones y un valor medio determinado. A continuación, se selecciona una de las variables explicativas y un punto de corte que permita generar la mayor diferencia entre los promedios de los dos nuevos segmentos o nodos. Una vez se obtiene la segmentación óptima se asigna un valor de predicción a los nodos que no tienen sucesores (hojas). Todas las observaciones que se encuentran en la misma hoja reciben el mismo valor de predicción para el caso en que la variable objetivo sea de intervalo y se denominan árboles de regresión, o la misma probabilidad de evento si la variable objetivo es binaria en cuyo caso se denominan árboles de clasificación. En la Figura 4 se muestra la estructura general de los árboles de decisión.

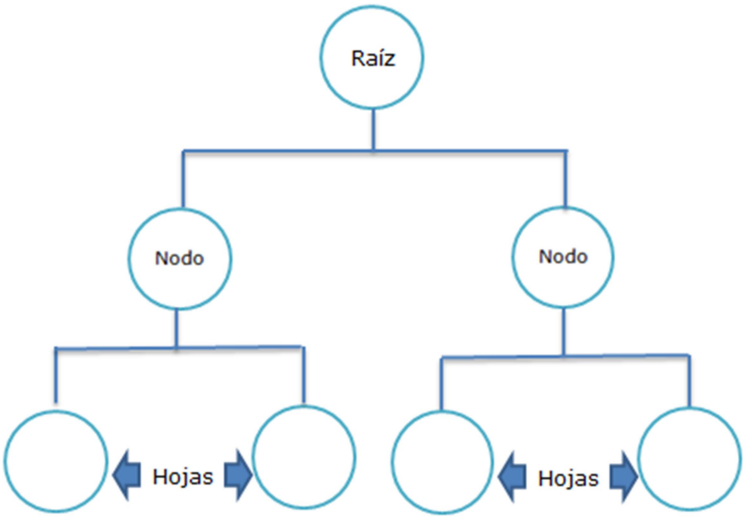


Figura 4. Estructura general árboles de decisión.

Los parámetros que se pueden configurar en un árbol son:

- Número de divisiones máximas en cada nodo. El más utilizado es el binario que divide en dos.
- Número de hojas o profundidad del árbol. Se puede utilizar poda para que el sistema evalué la cantidad de hojas que tendrá el árbol.
- P-valor: determina las divisiones en cada nodo. Entre más bajo, restringe más el número de subdivisiones y da lugar a arboles más sencillos. Esto conlleva a menor varianza a cambio de mayor sesgo en los modelos.
- Tamaño de la hoja, número de observaciones mínimo que debe de haber en un nodo. Se deben evitar hojas con pocas observaciones porque conllevaría a sobreajuste, es decir que el modelo se ajusta muy bien a los datos de entrenamiento y tiene un sesgo bajo, pero presenta una varianza muy alta frente a nuevos datos.

Las ventajas que tienen los árboles son:

- Adaptabilidad a la forma funcional entre variable objetivo y predictoras.
- Tratamiento automático de valores missings.
- Tratamiento automático de categorías poco representadas.
- Detección automática de regiones y puntos de corte (no tratada en otros algoritmos o técnicas).
- Resultados a menudo fáciles de comprender.

En cuanto a las desventajas se encuentra que tienen:

- Poca capacidad predictiva y gran varianza.
- Sensibilidad a cambios en los datos, inestabilidad y poca robustez.
- Falta de suavidad (función escalonada) lo que a veces redunda en mayor error promedio de predicción en regresión.

Estas desventajas no se han podido solucionar mejorando los algoritmos de construcción, pero sí combinando el resultado de muchos árboles como se verá en las técnicas que se describen a continuación.

3.6.1 Bagging (Bootstrap Averaging)

Es uno de los métodos básicos de ensamble o combinado de modelos, que consiste en promediar las predicciones obtenidas de una cantidad de árboles que se decidan construir, logrando así reducir la varianza del modelo aunque se sacrifica la interpretación del mismo. Esta técnica sortea las observaciones con o sin reemplazo para hacer m árboles o iteraciones. Se puede definir tanto la cantidad m de iteraciones como el tamaño de la muestra a utilizar y la complejidad de los arboles (número de hojas final o profundidad, número máximo de observaciones por nodo, número de divisiones máximo por nodo, etc.).

3.6.2 Random Forest (RF)

Es una generalización del bagging, adicionando la característica de sortear las variables a utilizar para segmentar cada nodo del árbol, consiguiendo arboles más variados. Adicional a las características del bagging, requiere que se defina el número de variables ***p*** a muestrear del total de variables independientes. Esta aleatoriedad en la selección de las variables evita la rigidez de utilizar un solo conjunto de variables y permite generar arboles más variados que logran una mejor adaptación a los datos, reduciendo a la vez el riesgo de sobreajuste porque no depende de un solo árbol sino que utiliza el promedio de muchos.

3.6.3 Incremento Gradiente

Este método, a diferencia de los anteriores, no promedia las predicciones de los árboles, sino que se basa en repetir los árboles, modificando levemente las predicciones iniciales intentando minimizar los residuos en la dirección de decrecimiento.

Los parámetros a definir son la constante de regularización que precisa en cuánto se modifica el error con cada iteración, el número de iteraciones y la configuración del árbol. Los algoritmos de esta técnica se describen en las Figuras 5 y 6.

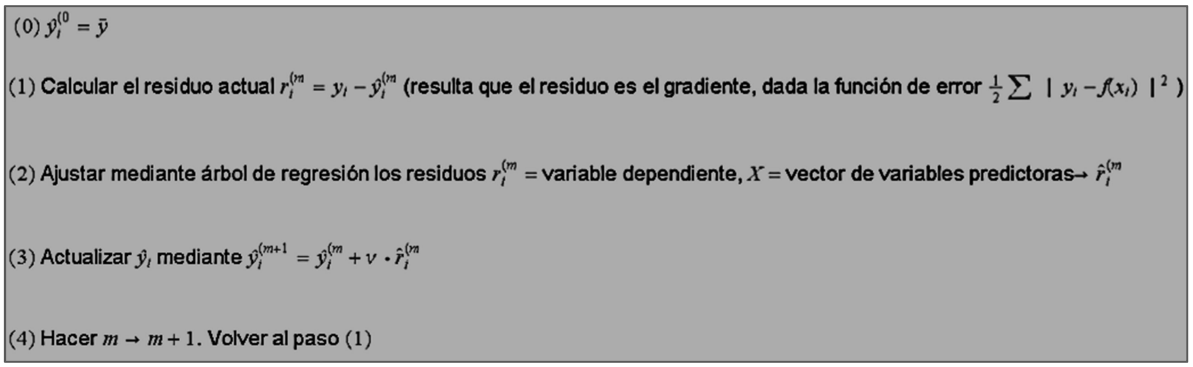


Figura 5. Algoritmo incremento gradiente para regresión. Fuente: Portela (2019)

Se repiten los pasos 1 a 4 hasta que converja o se presente sobreajuste.

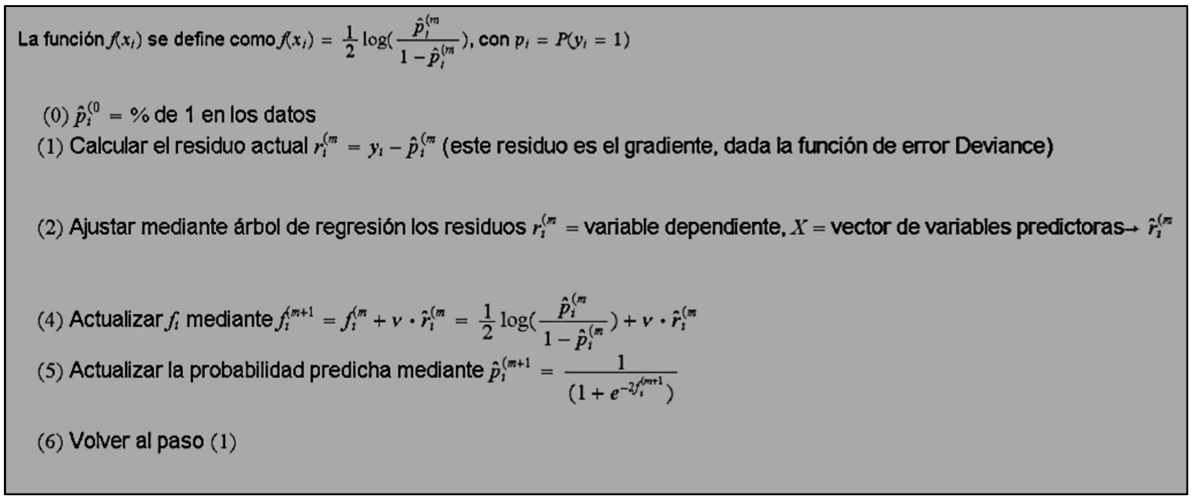


Figura 6. Algoritmo incremento gradiente para clasificación. Fuente: Portela (2019).

3.6.4 XGBoost

Es una modificación del incremento gradiente involucrando regularización o penalización en el momento de construir cada árbol. Esta función de penalización previene el sobreajuste y prefija dos prefija dos parámetros lambda y gamma ($\gamma\ y\ \lambda$).

3.7 Redes Neuronales

Las Redes Neuronales imitan el funcionamiento del cerebro humano para realizar tareas de aprendizaje. Tienen una arquitectura organizada en capas de neuronas, las cuales tienen pesos asignados a sus interconexiones. El aprendizaje de la red consiste en ajustar los pesos mediante una regla que indica cómo modificarlos en función de los datos de entrenamiento.

La capa input o de entrada compuesta por las X_i variables independientes, se conecta a la capa oculta H_j mediante la función de combinación representada en la Figura 7 con el símbolo Σ donde los pesos w_{ij} son los parámetros a estimar, la función de combinación más habitual es la lineal.

Después de aplicar la función de combinación se aplica a cada nodo oculto la función de activación representada en la Figura 7 por ∫. La función de activación más utilizada es la tangente hiperbólica. Se repite el proceso para todas las capas ocultas que tenga la red hasta llegar a la capa output.

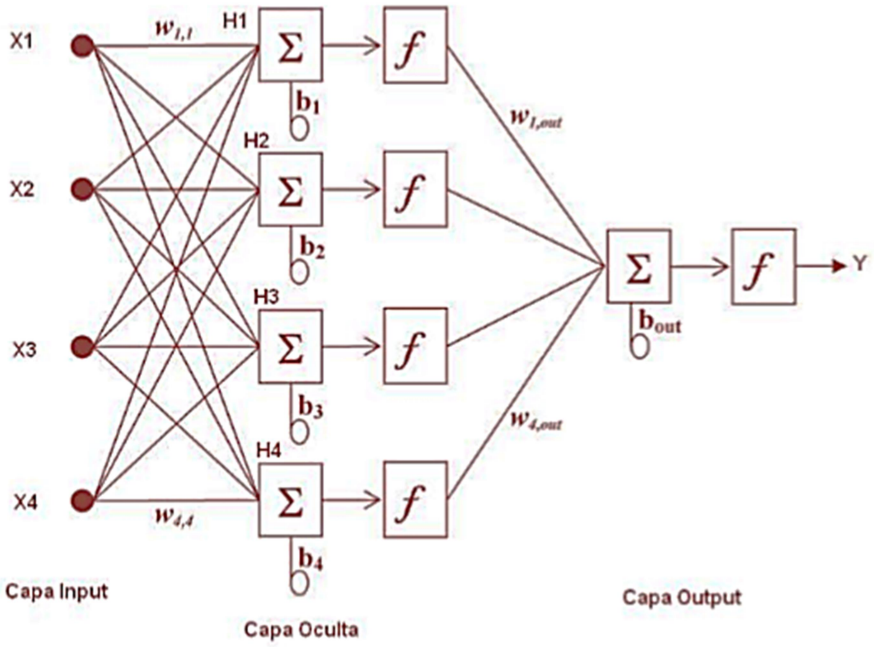


Figura 7. Estructura redes neuronales. Fuente: Portela (2019).

Para una red con tres variables independientes, una variable de salida, una capa oculta y dos nodos ocultos, donde el parámetro constante b_j se denomina sesgo, el modelo seria el siguiente:

$$Y = \tanh(w_1, out(\tanh(w_{11}X1+w_{21}X2+w_{31}X3 + b_1)) + w_2, out(\tanh(w_{12}X1+w_{22}X2+w_{32}X3 + b_2)) + b_{out})$$

El objetivo de la red neuronal es estimar los parámetros $\boldsymbol{w}$ y $\boldsymbol{b}$, utilizando métodos de optimización numérica que van variando los valores de los parámetros de forma iterativa hasta cumplir el objetivo de optimización. Para definir la configuración óptima de la red se deben evaluar el número de nodos, la función de activación, el número máximo de iteraciones y el algoritmo de optimización a utilizar.

3.8 Máquinas de soporte vectorial (SVM)

Utiliza la relación observada entre las variables independientes o predictoras y la variable dependiente o respuesta, para construir un hiperplano de separación que subdivide el conjunto de observaciones.

El objetivo de las máquinas de soporte vectorial es obtener el vector de parámetros W (que soportan la construcción de los hiperplanos), maximizando la distancia entre los dos hiperplanos de separación o margen, en la Figura 8, la línea negra continua representa el hiperplano óptimo de separación y el margen en líneas discontinuas. En esta figura hay tres observaciones (dos de la clase azul y una de la morada) que son equidistantes al hiperplano y que se encuentran sobre el margen y son conocidas como vectores soporte.

Para maximizar la distancia entre los hiperplanos se utilizan métodos clásicos de optimización permitiendo un margen de error en la separación (ε) y un número

máximo de observaciones que superen ese margen denominado constante de regularización (C).

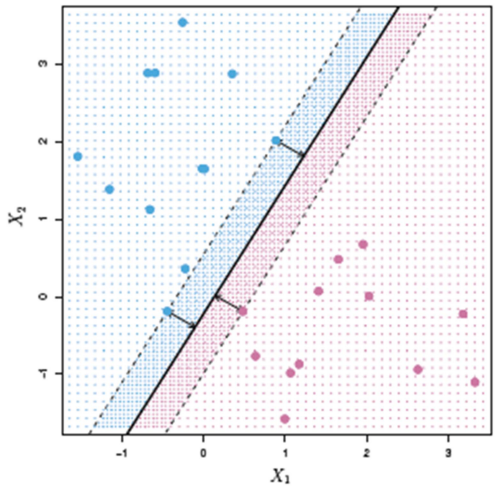


Figura 8. Máquinas de soporte vectorial. Fuente: Libro ISLR

En la mayoría de los casos, la separación entre clases no es lineal, por lo cual se debe trabajar en un espacio de dimensión superior donde tenga sentido la separación lineal. Como se puede observar en la Figura 9, al proyectar las observaciones se logra mayor separabilidad. La función de transformación puede ser, por ejemplo, de tipo Kernel.

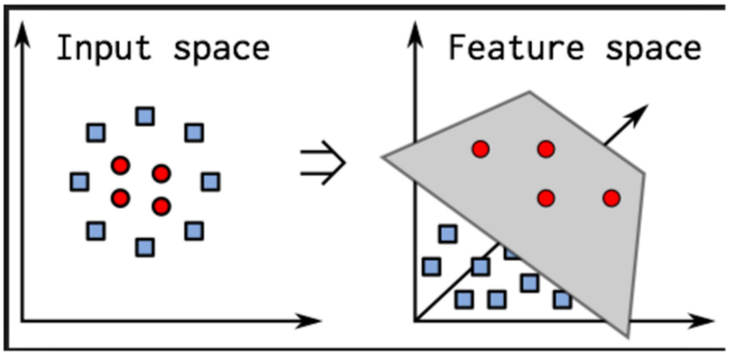


Figura 9. Proyección a una dimensión superior. Fuente: Portela (2019).

El Kernel que se utiliza con más frecuencia de los no lineales es el RBF Gaussiano, cuya función se puede ver en la Figura 10, este necesita un parámetro sigma o gamma. El Kernel polinomial necesita el grado del polinomio y el sigmoide, la posición y escala.

Linear function	$K(\mathbf{x}_i \cdot \mathbf{x}_j) = \mathbf{x}_i^T \cdot \mathbf{x}_j$
Polynomial function	$K(\mathbf{x}_i \cdot \mathbf{x}_j) = \left(\gamma \mathbf{x}_i^T \cdot \mathbf{x}_j + r\right)^d$
RBF function	$K(\mathbf{x}_i \cdot \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)$ $= \exp\left(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2\right), \gamma = \frac{1}{2\sigma^2}$
Sigmoid function	$K(\mathbf{x}_i \cdot \mathbf{x}_j) = \tanh\left(\gamma \mathbf{x}_i^T \cdot \mathbf{x}_j + r\right)$ $\tanh(x) = \frac{e^x}{e^x + 1}$

Figura 10. Funciones Kernel. Fuente: Portela (2019).

3.9 Ensamblado

Su objetivo es construir predicciones a partir de la combinación de varios modelos. Los ensambles suelen ser más fiables que un solo algoritmo, ya que unos modelos corrigen a los otros. Cuanto menor sea la correlación entre modelos, menor será el error global de ensamblado. Se trata de ensamblar modelos con sesgo similar.

Los métodos basados en arboles como bagging, random forest e incremento gradiente descritos anteriormente son un ejemplo de ensamblado.

Se utiliza en general el término ensamblado para cualquier tipo de combinación de modelos, existen tres opciones básicas:

1. **Promediado:** Se calcula el promedio de las predicciones, si los modelos son de clasificación se calcula el promedio de las probabilidades.
2. **Voto:** Esta opción aplica para clasificación, la predicción será aquella que se haya repetido más veces, por ejemplo: si la predicción del modelo 1 es cero, la predicción del modelo 2 es uno y la predicción del modelo 3 es cero. La predicción del ensamblado será cero.
3. **Combinación a partir de otro algoritmo**: Se introducen como variables de entrada las predicciones de los modelos a ensamblar.

4. Predicción del coste total anual del servicio de salud

4.1 Definición de la variable objetivo

El objetivo es predecir el coste total anual del servicio de salud de una entidad. La variable objetivo como se mencionó en la metodología, para el modelo de dos partes se dividirá en dos: coste_binario y coste_total. Para esto se divide la base de datos en dos, una con la totalidad de los datos con variable objetivo binaria y otra que contenga exclusivamente los valores diferentes de cero. Adicionalmente, se utilizarán la totalidad de los datos y se predecirá con modelos clásicos el coste total para finalmente comparar los resultados.

El número de observaciones de la clase de interés (1 tiene coste) es de 4447 sobre 10331 que corresponde al 43% del total de observaciones. Esto significa que si no se realiza ningún modelo y se predijera que todas las observaciones son cero (0 no coste), el error sería del 43% (tasa de fallo). Teniendo en cuenta lo que puede significar un error de esta magnitud en términos monetarios, cualquier mejora en este indicador representará contar con las reservas presupuestales necesarias para que la empresa no presente déficit financiero. Se verificará que el modelo ganador mejore está medida de error.

4.2 Muestreo de datos

Los datos fueron obtenidos a través de un convenio de colaboración con una EPS colombiana. La entidad cuenta actualmente con más de 2 millones de afiliados, se utilizará una muestra aleatoria de 10331 registros que corresponden a las personas que estuvieron afiliadas a la EPS durante un año tomados al corte de diciembre 31. Las variables entregadas por la entidad se encuentran en la Tabla 1.

Tabla 1. Listado de variables entregadas por la entidad.

Variable	Descripción
id_afiliado	identificación del usuario
Zona	código ubicación o zona geográfica en la que se encuentra el afiliado
cantidad_facturaciones	cantidad de facturaciones en un año por usuario
cantidad_servicios	cantidad total de los servicios incluidos por todas las facturas en un año por usuario
Coste (Variable Objetivo)	Suma agregada del coste total en un período de un año calendario por la modalidad de contratación evento (Coste variable), para la cobertura PBS.
estado_afiliado	Si es del régimen contributivo o subsidiado (movilidad descendente).
Genero	Genero del usuario masculino o femenino
Edad	edad del usuario
tipo_cotizante	código que identifica el tipo de cotizante: Cotizante, Beneficiario o segundo cotizante.
codigo_cohortes	Existen 9 cohortes enumerados del 1 al 9. Si un usuario presenta un número de un solo digito, ejemplo 8, quiere decir que hace parte de la cohorte 8, pero si presenta un número de dos dígitos, por ejemplo 26, quiere decir que pertenece a las cohortes 2 y 6. Las cohortes son los grupos de riesgo definidos por la entidad para los diagnósticos de alto coste.
dias_afiliacion	Cantidad de días que el usuario lleva afiliado a la EPS a la fecha final del período analizado.

4.3 Exploración de los datos

Se realiza un análisis descriptivo de los datos para detectar tanto posibles relaciones entre las variables explicativas y la variable objetivo, así como inconsistencias en la información.

En cuanto a la variable input "código cohortes", en la Tabla 2 se relaciona el código y la descripción. Para esta variable se crearán las nueve variables individuales de tipo binaria, que permita clasificar a los afiliados de acuerdo con la/las enfermedades reportadas, el nombre de las variables a crear coincide con la descripción de la enfermedad de la Tabla 2. Por ejemplo, si una persona requiere diálisis tendrá un valor de 1 en la variable Diálisis del modelo.

Como se puede observar en la Tabla 3, las variables categóricas no tienen datos ausentes, en cuanto al número de niveles, por ser la mayoría variables binarias, presentan dos niveles con excepción de zona que tiene 8, para esta variable se crearán siete variables dummies.

Tabla 2. Descripción del código cohorte.

Código Cohorte	Descripción
1	Diálisis
2	Enfermedades huérfanas
3	Esclerosis
4	Fibrosis quística
5	Hemofilia
6	Oncología adultos
7	Oncología pediátrica
8	Reumatología y enfermedades del colágeno
9	VIH

Tabla 3. Exploración de datos – variables categóricas

Variable	Tipo	Niveles	Ausente
zona	N	8	0
VIH	N	2	0
tipo_cotizante	C	3	0
sexo	C	2	0
Reumatologia_colageno	N	2	0
Oncologia_adultos	N	2	0
estado_afiliado	C	2	0
Dialisis	N	2	0

En cuanto a las variables de intervalo, al revisar los estadísticos en la Tabla 4, se puede observar que las variables presentan valores normales de asimetría, máximos y mínimos, no se evidencian datos atípicos, ni valores ausentes. Cabe destacar que la variable días afiliación tiene una desviación alta, analizando la frecuencia en la Figura 11, se puede observar que la mayoría llevan menos de 2 años afiliados a la entidad y existen pocos afiliados con una antigüedad superior a 7 años, esta variable presenta una asimetría positiva.

Tabla 4. Exploración de datos – variables intervalo

Variable	Mínimo	Máximo	Media	Mediana	Desv. Est.	Asimetría	Curtosis	Ausente
dias_afiliacion	1	8545	2322.6568	1371	2257.9191	1.0563499	-0.012371	0
edad	1	102	34.65758	32	21.046809	0.3283912	-0.586999	0

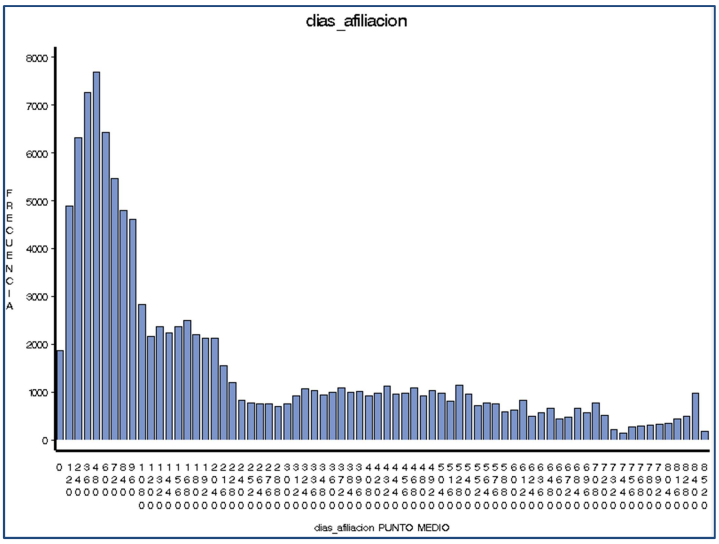


Figura 11. Días afiliación.

En cuanto al género, existe una mayor proporción de mujeres que de hombres con un 57,5%. Analizando el gráfico de la Figura 12 género por rango de edad, se observa que de estas más del 50% se encuentra entre los 21 y 50 años. De acuerdo con el tipo de cotizante de la Figura 13, las mujeres en su mayoría son beneficiarias del sistema mientras que los hombres tienen una mayor participación como cotizantes.

En la Figura 14, el número de afiliados por zona muestra que la proporción de hombres y mujeres esta equilibrada, se destaca que la zona 7 tiene una cantidad mínima de afiliados, mientras que la zona 2 cuenta con el 65% del total.

Por último, en la Figura 15, se observa que el 19% de los usuarios presenta menos de un año de afiliación por lo cual el valor registrado de coste anual corresponde a un valor fraccional, se utilizará la variable días afiliación porcentaje para evidenciar esta característica.

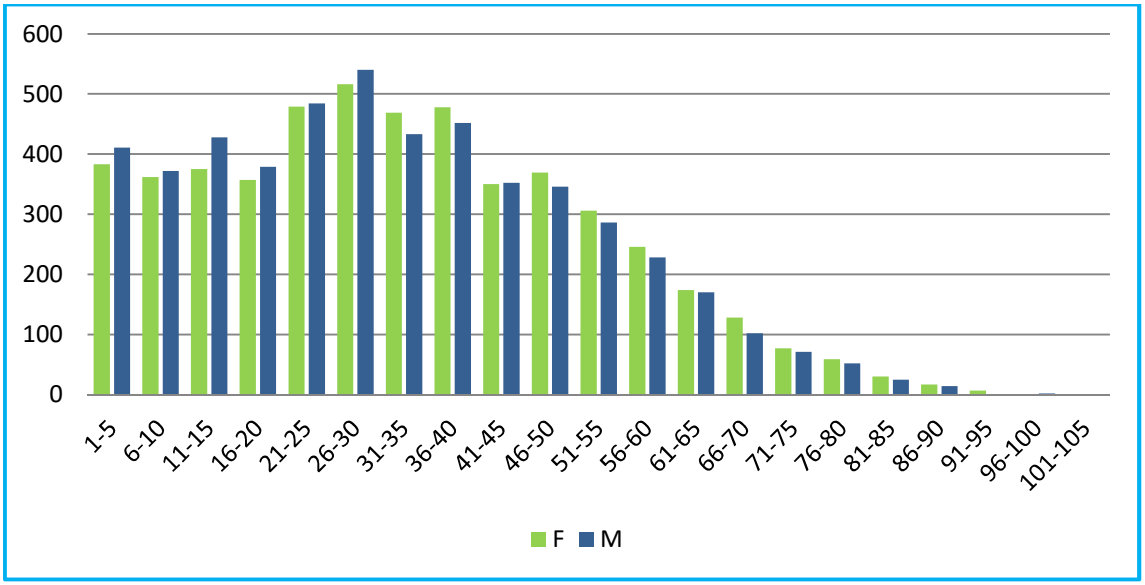


Figura 12. Número de afiliados por género y rango de edad

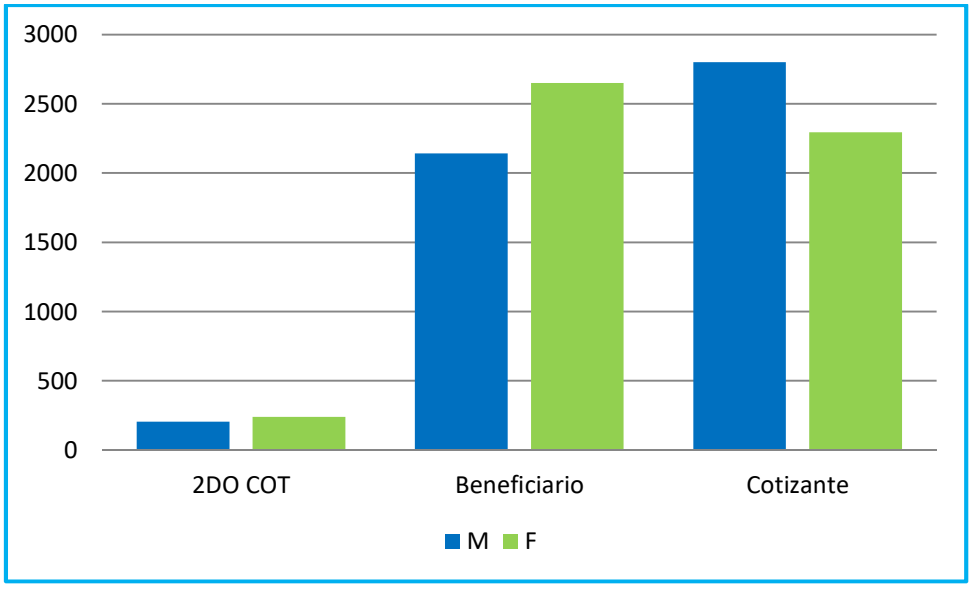


Figura 13. Número de afiliados por tipo de cotizante y género.

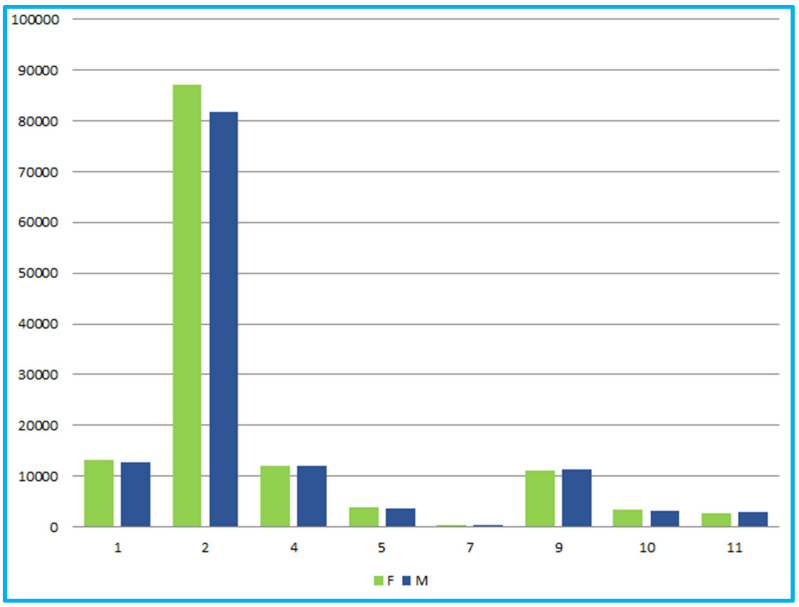


Figura 14. Afiliados por zona y género

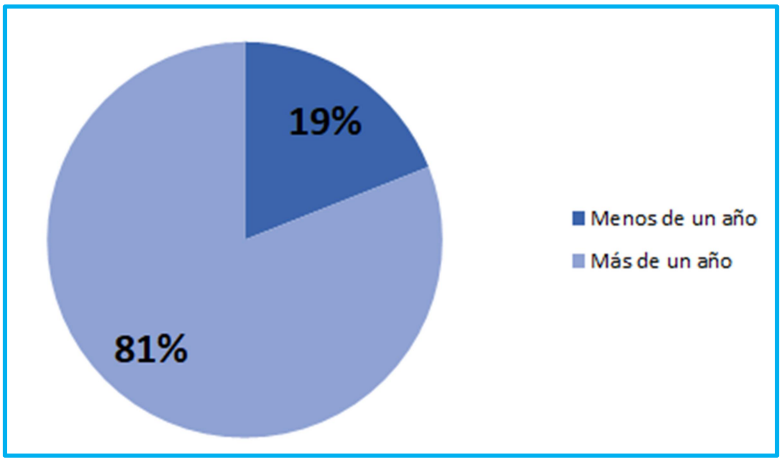


Figura 15. Tiempo de afiliación.

En cuanto a la variable objetivo como se puede ver en la Figura 16, cumple las dos características que se requieren para aplicar un modelo de dos partes, tiene una alta proporción de valores cero, que indican que el 57% de los afiliados no utilizaron el servicio de salud en el periodo analizado, y presenta una fuerte asimetría positiva, esto es debido a que la mayoría de las personas que requieren el servicio de salud presentan costes bajos y solo una pequeña proporción genera costes elevados.

Al analizar el coste anual por rangos de edad y genero de los afiliados, como puede observarse en la Figura 17, las mujeres generalmente tienen un mayor coste que los hombres, excepto en los rangos de 76 a 85 años. A nivel general, el coste presenta un comportamiento cuadrático en función de la edad, por esta razón se incluirá la variable $edad^2$.

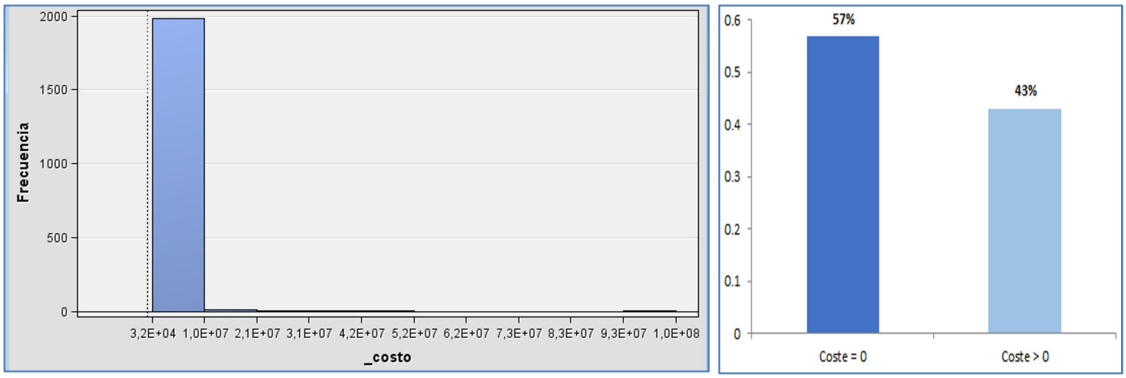


Figura 16. Características del coste anual frecuencia (izquierda) y porcentaje de coste igual a cero y mayor que cero (derecha).

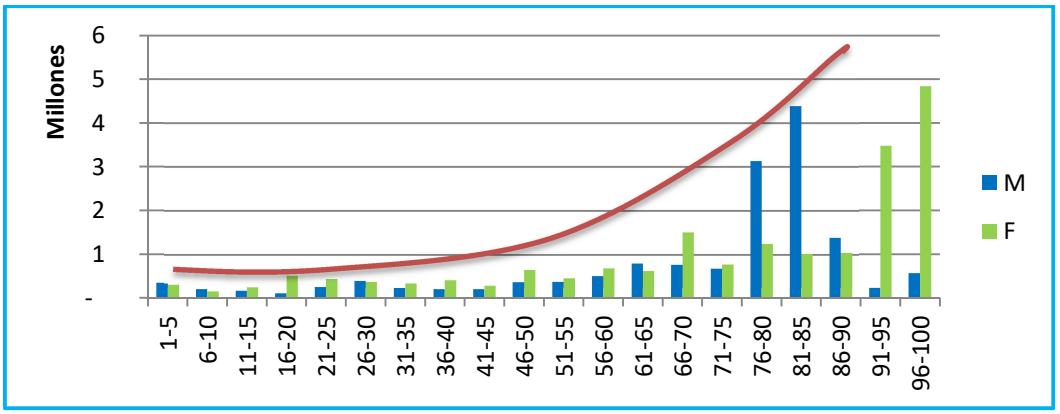


Figura 17. Coste medio anual por rangos de edad y género.

4.4 Modificación

4.4.1 Reducción inicial de variables

Las variables Cantidad_Facturaciones que hace referencia a la cantidad de facturas que se emiten al afiliado y Cantidad_Servicios utilizados, no serán incluidas en el modelo porque no se conocen a priori y por lo tanto no sirven para predecir.

Analizando la cantidad de observaciones distintas de cero por variable binaria, se descartaron aquellas que tengan un porcentaje inferior al 0.05%, por considerarse

no representativas de las características de la población en estudio. Estas son: Enfermedades huérfanas, Esclerosis, Fibrosis quística, Hemofilia y Oncología pediátrica.

4.4.2 Creación de nuevas variables

Se crea una nueva variable "enfermedades totales" que suma el número de enfermedades que presenta un mismo afiliado, siendo cero (ninguna), uno (una enfermedad) y dos (dos o más enfermedades). Esta variable servirá para verificar si existe alguna relación entre tener más de una enfermedad y la generación del coste.

En cuanto al tiempo, tomando como base la variable días de afiliación, se crean dos nuevas variables. Una variable que identifican si el afiliado lleva menos de un año y otra variable que relaciona el porcentaje del último año que ha estado afiliado, si la persona lleva un año o más afiliado este porcentaje tomará el valor de 100%, en caso contrario se calculará de acuerdo con la cantidad de días que ha estado afiliado sobre 365 días.

Como se identificó previamente, existe una relación cuadrática entre la edad y el coste, por lo cual se crea la variable edad2 = $edad^2$. Adicionalmente, se evidencia variaciones en el coste dependiendo de la edad y el género, por esta razón se crean las variables de interacción de edad con género.

Por medio del nodo selección de variables de SAS Enterprise Miner ®, se agrupan las variables: edad, días afiliación, edad2, enfermedades totales, estado afiliado y tipo de cotizante. Así mismo, por medio del nodo transformación de variables del mismo programa se crean las variables de interacción de edad con género.

Todas las variables categóricas se convierten a dummies y se utilizan (K-1) variables binarias para cada una de ellas. Finalmente, se obtienen un total de 36 variables explicativas, seis de intervalo, 30 binarias y una identificativa, como se puede observar en la Tabla 5.

Tabla 5. Listado de variables independientes o explicativas.

variable	Descripción	Tipo
id_afiliado	Identificación del usuario	Identificativa
edad	Edad del afiliado	Intervalo
dias_afiliacion	Cantidad de días que el usuario está afiliado a la EPS a la fecha final del período analizado 31-dic	Intervalo
dias_afil_porc	Porcentaje de días que ha estado afiliado en el año actual.	Intervalo
menos1	Personas que tienen menos de un año de afiliación	Binaria
dialisis	Personas que requieren servicio de diálisis	Binaria
oncologia_adultos	Personas que presentan cáncer	Binaria
reumatologia_colageno	Personas con enfermedades relacionadas con reumatología y enfermedades del colágeno	Binaria
VIH	Personas con VIH	Binaria

zona_1	Personas que se encuentran afiliadas a la zona 1	Binaria
zona_2	Personas que se encuentran afiliadas a la zona 2	Binaria
zona_4	Personas que se encuentran afiliadas a la zona 4	Binaria
zona_5	Personas que se encuentran afiliadas a la zona 5	Binaria
zona_9	Personas que se encuentran afiliadas a la zona 9	Binaria
zona_10	Personas que se encuentran afiliadas a la zona 10	Binaria
zona_11	Personas que se encuentran afiliadas a la zona 11	Binaria
edad2	Edad elevada al cuadrado	Intervalo
edad_F	Interacción de la edad con el género Femenino	Intervalo
edad_M	Interacción de la edad con el género Masculino	Intervalo
TI_edad21	Agrupación de edad2 menor que 20,5 y mayor que 3192,5	Binaria
TI_edad22	Agrupación de edad2 con valores entre 420,5 y 3192,5	Binaria
TI_G_enf_totales1	Agrupación de personas que presentan 1 o más enfermedades	Binaria
TI_dias_afil1	Rango de días afiliación menores de 297,5	Binaria
TI_dias_afil2	Rango de días afiliación entre 297,5 y 8295	Binaria
TI_OPT_edad21	Rango de edad2 dato menor de 20,5	Binaria
TI_OPT_edad22	Rango de edad2 entre 20,5 y 420,5	Binaria
TI_OPT_edad24	Rango de edad2 superior a 3192,5	Binaria
TI_enf_totales1	Personas que no tienen registrada ninguna de las enfermedades	Binaria
TI_enf_totales2	Personas que tienen registrada una de las enfermedades de alto costo	Binaria
TI_estado_afiliado1	Personas que su estado de afiliado es activo en el régimen	Binaria
genero_F	Género Femenino	Binaria
TI_tipo1	Tipo de cotizante: segundo cotizante (2DO COT)	Binaria
TI_tipo2	Tipo de cotizante: Beneficiario	Binaria
TI_OPT_edad1	Agrupación de los afiliados entre 1 y 5 años	Binaria
TI_OPT_edad2	Agrupación de los afiliados entre 6 y 21 años	Binaria
TI_OPT_edad3	Agrupación de los afiliados entre 22 y 64 años	Binaria
TI_OPT_edad4	Agrupación de los afiliados mayores de 65 años	Binaria

4.4.3 Selección de variables

Algunas de las técnicas como la regresión y árboles tienen sus propios criterios de selección de variables. Para evitar que las demás técnicas estén en desventaja, y/o utilizar variables que no aporten o sobreajusten el modelo, se realizan tres procesos previos de selección de variables: selección de variables a través de

diferentes técnicas en SAS Enterprise Miner ®, selección de acuerdo con la importancia de la variable y selección aleatoria en SAS Base por medio de la macro RandomSelect. A continuación se explica cada uno de ellos:

Selección Miner: Este proceso consiste en utilizar las técnicas que tienen su propia selección y realizar un conteo de la cantidad de veces que es utilizada cada variable. Las técnicas que se utilizan son: árbol, incremento gradiente, regresión con método de selección stepwise, backward y forward. Estas pruebas se realizan en el software SAS Enterprise Miner ®. También se utiliza el nodo de selección de variables que se encuentra disponible en esta herramienta para un total de 6 métodos de selección. Se realiza este proceso para las dos partes que se van a modelar con variable objetivo binaria y continua, En las Tablas 6 y 7 se muestran las variables que obtuvieron más de tres en el conteo:

Tabla 6. Resultado variables seleccionadas por seis métodos diferentes (Binaria)

Variable	Conteo
TI_edad21	5
TI_G_enf_totales1	5
edad_F	4
TI_dias_afil1	4
zona_9	4
edad	3
genero_F	3
TI_estado_afiliado1	3
TI_OPT_edad2	3
TI_OPT_edad3	3
TI_tipo2	3
zona_2	3
zona_4	3
zona_5	3

Tabla 7. Resultado variables seleccionadas por seis métodos diferentes (Continua)

Variable	Conteo
edad2	5
dialisis	3
edad	3
edad_F	3
oncologia_adultos	3
TI_G_enf_totales1	3
TI_OPT_edad22	3
TI_OPT_edad4	3
TI_tipo2	3
zona_2	3

Importancia de la variable: Otro conjunto de variables que se probará en el presente trabajo se obtiene utilizando la herramienta explorador de estadísticos de SAS Enterprise Miner ®, Validando la utilidad de todas las variables listadas en la Tabla 5, por medio del estadístico valor de la variable. Este estadístico corresponde

con –log (Pvalor), dado que el Pvalor indica como de probable es encontrar un valor más grande interesan Pvalores pequeños, por lo tanto el valor de la variable cuanto más grande sea será mejor. A priori se creó una variable aleatoria como punto de validación, todas las variables que tengan un valor similar o inferior serán descartadas, ya que no aportan información al modelo. Este conjunto se llamará "Importancia" de ahora en adelante.

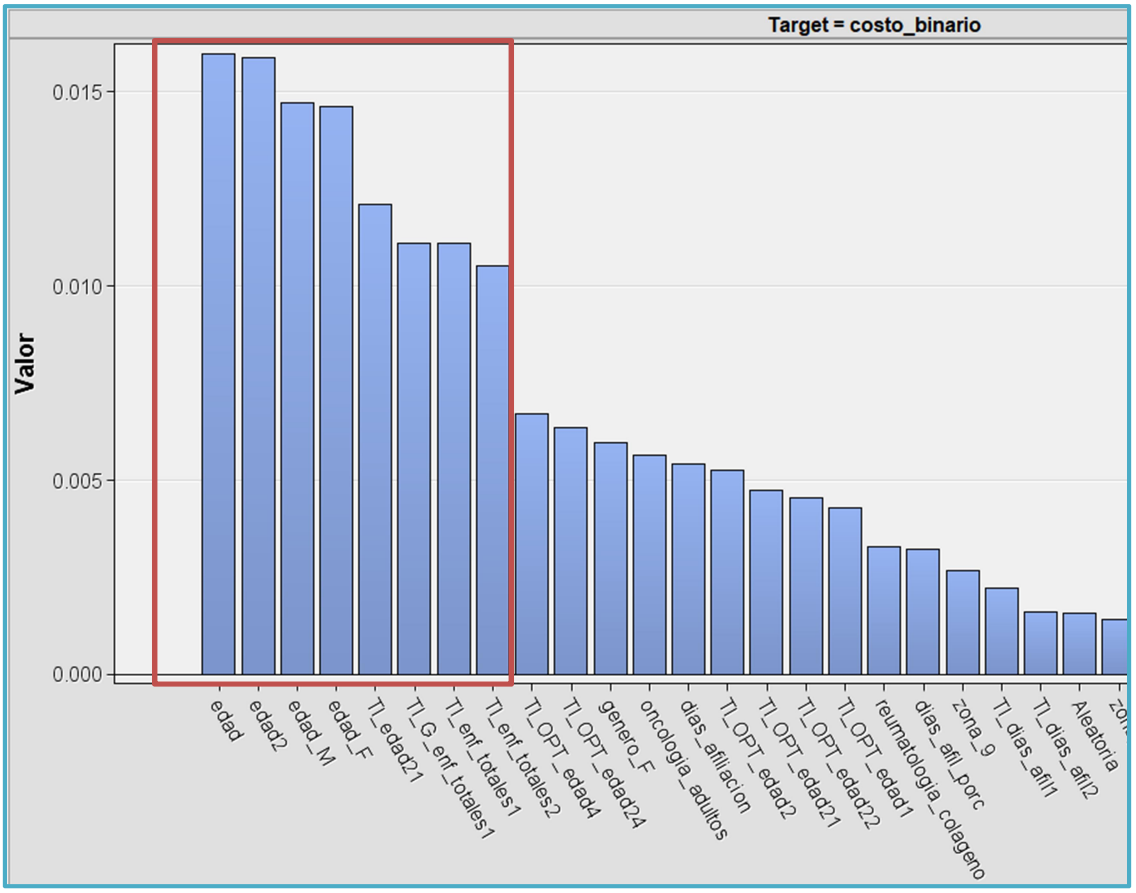


Figura 18. Valor de la variable para variable objetivo binaria

Se descartan los valores inferiores a 0.01 por considerarse que aportan muy poca información al modelo. De acuerdo con el valor de la variable, las variables seleccionadas se muestran en la Tabla 8.

Tabla 8. Variables seleccionadas para variable objetivo Binaria.

Variable	Importancia	Valor
edad	1	0.0160
edad2	2	0.0159
edad_M	3	0.0147
edad_F	4	0.0146
TI_edad21	5	0.0121
TI_G_enf_totales1	6	0.0111
TI_enf_totales1	7	0.0111
TI_enf_totales2	8	0.0105

Observando el gráfico de la Figura 19, podría decirse que las primeras cuatro variables son las que más aportan al modelo, en esta selección se incluirán las variables que se muestran en la Tabla 9, que a pesar de tener una importancia menor puede que aporten información al modelo.

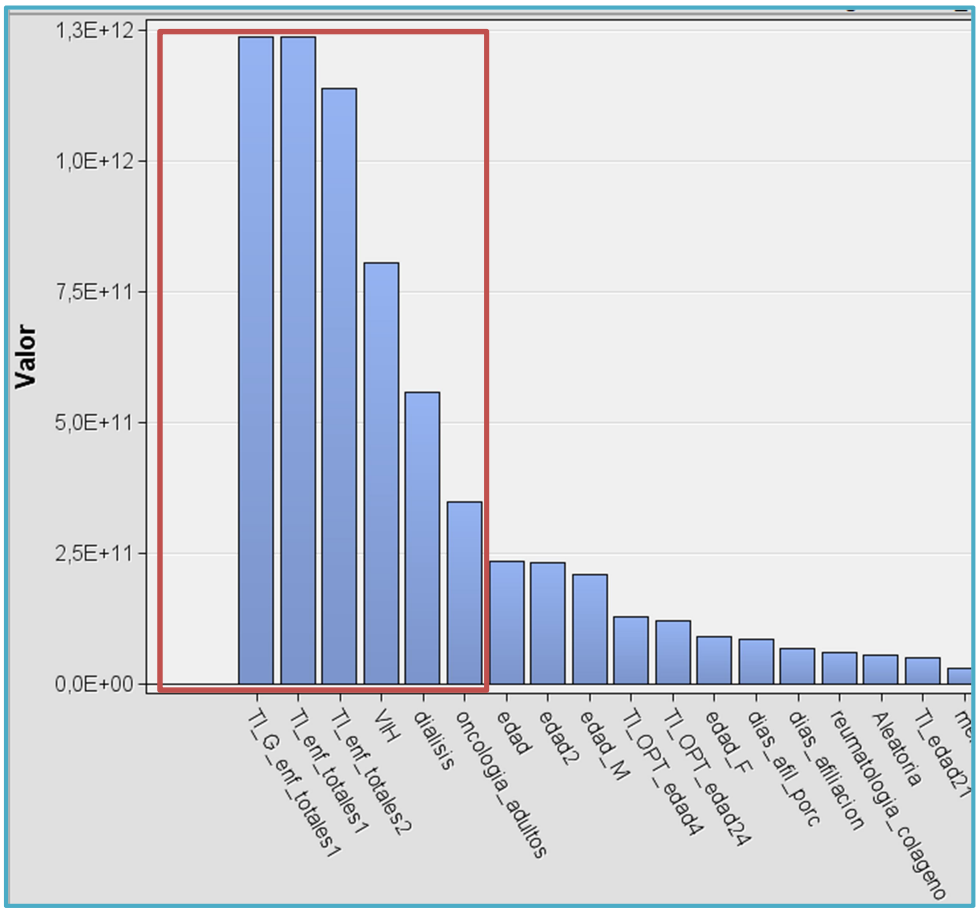


Figura 19. Valor de la variable para variable objetivo continua

Tabla 9. Variables seleccionadas para variable objetivo continua.

Variable	Importancia	Valor
TI_G_enf_totales1	1	1.24E+12
TI_enf_totales1	2	1.24E+12
TI_enf_totales2	3	1.14E+12
VIH	4	8.05E+11
dialisis	5	5.56E+11
oncologia_adultos	6	3.46E+11

Selección aleatoria: Para encontrar otros conjuntos de variables a probar se utiliza además la macro RandomSelect de Portela (2019), que realiza una selección de variables utilizando el criterio stepwise, y aplicando remuestreo variando la semilla 200 veces con un porcentaje de entrenamiento del 70%.

La salida arroja los conjuntos de variables utilizadas y el número de veces que fue utilizado este conjunto. Se seleccionaran aquellos conjuntos de variables que hayan sido seleccionados la mayor cantidad de veces. Como se puede observar en la

Tabla 10, nueve veces es el número de veces que más se repite un conjunto de datos que equivale al 4.6%, esto indica que no se tiene una estabilidad en las variables a utilizar. Para realizar pruebas se seleccionan los tres primeros conjuntos.

Tabla 10. Variables por selección aleatoria para variable objetivo Binaria

efecto	conteo	porcentaje
dias_afiliacion TI_edad21 TI_G_enf_totales1 TI_dias_afil1 TI_dias_afil2 TI_estado_afiliado1 genero_F TI_tipo2 zona_1 zona_4 zona_5 zona_9 TI_OPT_edad1 TI_OPT_edad4	9	4.66321
dias_afil_porc TI_edad21 TI_edad22 TI_G_enf_totales1 TI_estado_afiliado1 genero_F TI_tipo2 zona_1 zona_4 zona_5 zona_9 TI_OPT_edad1 TI_OPT_edad4	6	3.10881
dias_afil_porc TI_edad21 TI_edad22 TI_G_enf_totales1 genero_F TI_tipo2 zona_1 zona_4 zona_5 zona_9 TI_OPT_edad1 TI_OPT_edad4	6	3.10881

Igualmente para la variable objetivo continua, se utiliza la macro RandomSelect con los métodos stepwise, backward y criterios de selección AIC y BIC, esta macro utiliza el procedimiento glmselect, los resultados se pueden ver en la Tabla 11.

Tabla 11. Variables por selección aleatoria para variable objetivo continua.

efecto	conteo	porcentaje
Intercept dias_afiliacion menos1 dialisis oncologia_adultos reumatologia_colagen VIH edad2 TI_enf_totales1 TI_enf_totales2 TI_OPT_edad3	10	4.97512
Intercept dias_afiliacion menos1 dialisis oncologia_adultos reumatologia_colagen edad2 TI_enf_totales1 TI_enf_totales2 TI_OPT_edad4	5	2.48756

Este modelo al igual que el binario, la cantidad máxima de repeticiones es 10, se realizan pruebas con los dos primeros conjuntos que presentan la mayor frecuencia.

4.5 Modelo de dos partes Variable objetivo binaria (Parte 1)

Todas las funciones y macros de validación cruzada repetida, utilizadas en este trabajo fueron proporcionadas por Portela (2019) tanto para el software R como para SAS Base.

4.5.1 Regresión Logística

Se modelan los conjuntos de variables seleccionados para la primera parte, a través de la macro CruzadaLogistica. Para evitar el efecto de la aleatoriedad se evalúan con validación cruzada repetida para cuatro grupos y 200 semillas. La Tabla 12 contiene el listado de los conjuntos de variables seleccionados previamente:

Tabla 12. Conjuntos de variables seleccionadas para primera parte.

Conjunto 1	Conjunto 2	Conjunto 3	Conjunto 4	Conjunto 5
Selección Miner	**Importancia de la variable**	**Selección aleatoria 1**	**Selección aleatoria 2**	**Mejor con 10**
edad_F	edad	dias_afiliacion	dias_afiliacion	edad_F
edad	edad2	dias_afil_porc	dias_afil_porc	edad_M
TI_dias_afil1	edad_M	TI_edad21	TI_edad21	edad2
genero_F	edad_F	TI_G_enf_totales1	TI_edad22	TI_G_enf_totales1
TI_tipo2	TI_edad21	TI_dias_afil1	TI_G_enf_totales1	TI_dias_afil1
TI_estado_afiliado1	TI_G_enf_totales1	TI_dias_afil2	TI_estado_afiliado1	TI_tipo2
TI_edad21	TI_enf_totales1	TI_estado_afiliado1	genero_F	zona_1
TI_G_enf_totales1	TI_enf_totales2	genero_F	TI_tipo2	zona_5
TI_OPT_edad2		TI_tipo2	zona_1	zona_9
TI_OPT_edad3		zona_1	zona_4	TI_OPT_edad2
zona_9		zona_4	zona_5	
zona_2		zona_5	zona_9	
zona_4		zona_9	TI_OPT_edad1	
zona_5		TI_OPT_edad1	TI_OPT_edad4	
		TI_OPT_edad4		

Adicional a estos conjuntos, se prueban los mejores conjuntos seleccionados en SAS base con el procedimiento (proc logistic), desde seis hasta 10 variables. Para más detalle de las variables incluidas en cada conjunto, ver Anexo I. En la Tabla 13 se relacionan los modelos probados y sus resultados.

Tabla 13. Configuración modelos regresión logística parte 1.

Modelo	Conjunto de variables	Tasa de fallos	AUC	#Variables
reg1	Selección miner	0.3750	0.6460	14
reg2	Importancia	0.3850	0.6150	8
reg3	Selección aleatoria 1	0.3730	0.6460	15
reg4	Selección aleatoria 2	0.3750	0.6450	14
reg5	Mejor con 6 variables	0.3830	0.6330	6
reg6	Mejor con 7 variables	0.3770	0.6360	7
reg7	Mejor con 8 Variables	0.3740	0.6410	8
reg8	Mejor con 9 variables	0.3730	0.6430	9
reg9	Mejor con 10 variables	0.3720	0.6450	10

Las Figuras 20 y 21 muestran los resultados en R de las regresiones con los conjuntos probados. Como se puede observar, el conjunto de variables que presenta el menor sesgo es el reg9 (conjunto de variables "mejor con 10 variables"), compuesto por 10 variables, tres continuas y siete categóricas, en cuanto a los datos del área bajo la curva ROC que se puede observar en la Figura 21, la diferencia es mínima con respecto a reg3 que presenta el valor más alto y tiene cinco variables más. Se confirma que el mejor modelo con un valor medio de AUC de 0.645 y tasa de fallos del 37,2%. Igualmente en la Figura 22, se pueden ver los resultados en SAS Base confirmando que el mejor modelo se obtiene con este conjunto de variables.

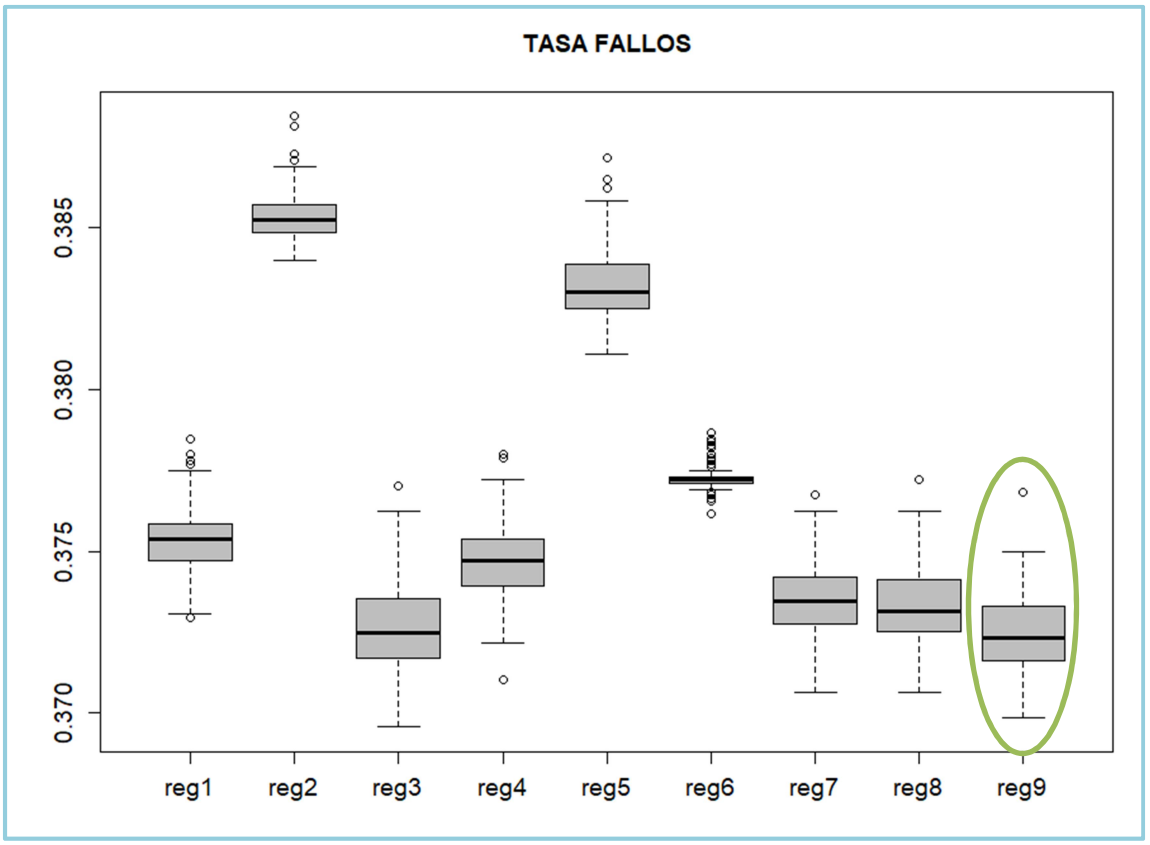


Figura 20. Resultados regresión logística en R, tasa de fallos parte 1.

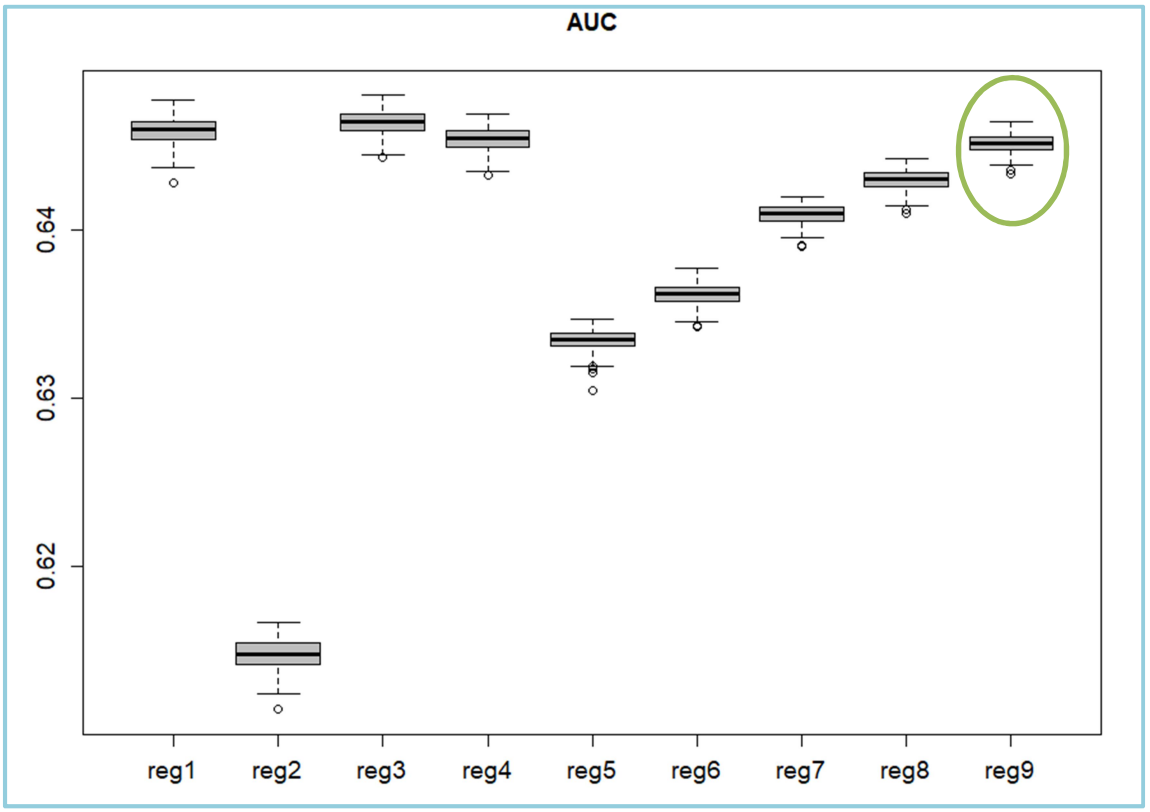


Figura 21. Resultados regresión logística en R, AUC parte 1.

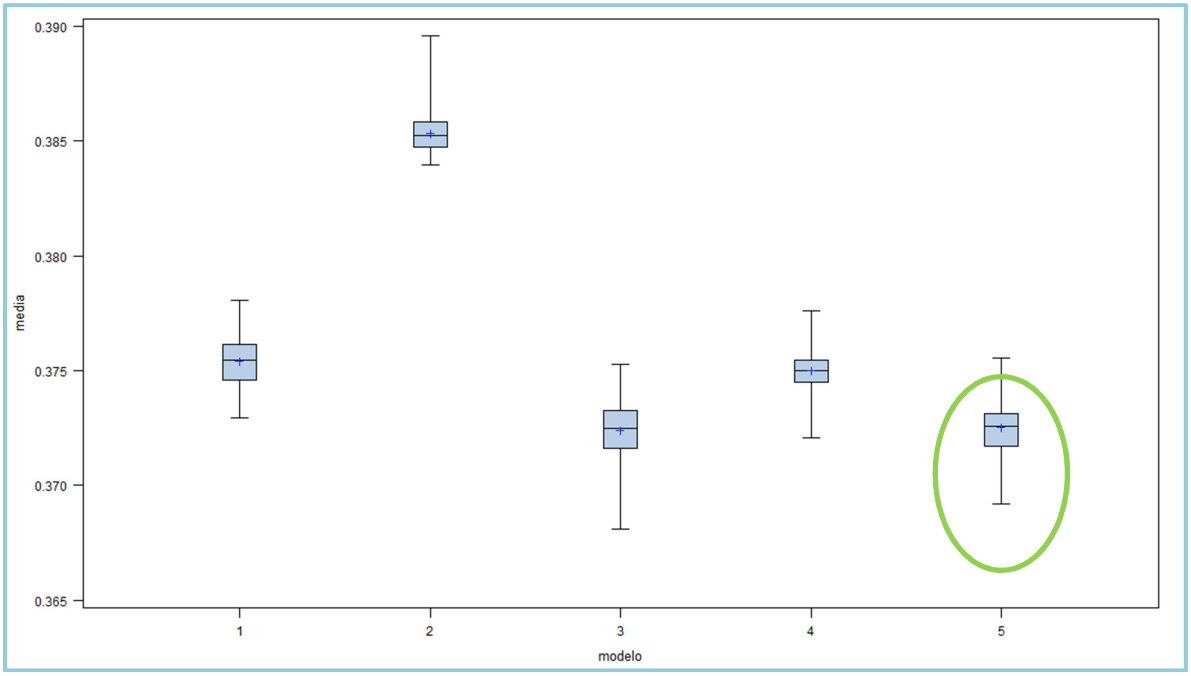


Figura 22. Resultados regresión logística SAS Base parte 1. - 1 selección miner, 2 importancia, 3 selección aleatoria 1, 4 selección aleatoria 2 y 5 mejor con 10.

4.5.2 Redes neuronales clasificación

Las redes neuronales requieren de muchas pruebas para encontrar una buena configuración, y aun así, no se garantiza que sea la óptima. Los principales parámetros a controlar son el número de nodos, la función de activación, el número máximo de iteraciones y el algoritmo de optimización.

El número de nodos que se pueden utilizar en una red dependen en cierta medida de la cantidad de observaciones con que se cuente, debido a que se tienen 4.447 observaciones de la variable de interés (cuando la variable objetivo toma valor 1), si se utiliza el 70% de los datos para entrenamiento se tendrían 3558 registros, aplicando la regla de 20 observaciones mínimas por nodo se tendría que:

$$\#\frac{\min obs}{parametro} = h(k+1) + h + 1 \;\; donde\; h = \#nodos\; ocultos\; y\; k = \#\; variables\; independientes$$

El número mínimo de observaciones por parámetro seria = 3558/20 = 178. Se calcula el número de nodos máximo a utilizar para cada conjunto, en la Tabla 14 se muestran los resultados.

Tabla 14. Nodos calculados para cada conjunto de variables parte 1.

Conjunto de variables	# Variables continuas	# Variables categoricas	# Variables totales	# Nodos calculados
Selección Miner	2	12	14	11
Importancia de la variable	4	4	8	18
Aleatoria 1	2	13	15	10
Aleatoria 2	2	12	14	11
Mejor con 10	3	7	10	15

Para definir el número de nodos a utilizar en la red, se realizan pruebas con validación cruzada repetida con diferentes valores iniciando en 3 como mínimo, incrementando dos nodos hasta llegar al máximo que coincide con el número calculado para cada conjunto. Utilizando la función Caret de R, que permite probar diferentes valores para los nodos (size) y la tasa de aprendizaje (decay) de 0.01, 0.1 y 0.001, con un máximo número de iteraciones de 200 y 10 repeticiones. También se prueba con SAS Base, el número de nodos con la macro variar (ver Anexo III para más detalle), los resultados de las dos pruebas se pueden ver en la Tabla 15.

Tabla 15. Número de nodos y tasa de aprendizaje a probar parte 1.

Conjunto de variables	Nombre	#Nodos R	Tasa de aprendizaje (decay)	#Nodos SAS
Conjunto 1	Selección Miner	3	0.01	3
Conjunto 2	Importancia	3	0.001	5
Conjunto 3	Aleatoria 1	3	0.1	3
Conjunto 4	Aleatoria 2	3	0.1	3 o 7
Conjunto 5	Mejor con 10	3	0.001	3

Una vez definido el número de nodos y la tasa de aprendizaje, se prueban en R las redes con validación cruzada repetida 200 veces utilizando la función cruzadaavnnetbin, las configuraciones probadas y sus resultados se muestran en la Tabla 16.

Tabla 16. Configuración y resultados redes en R parte 1.

Modelo	Conjunto de variables	# Nodos	Tasa de aprendizaje (decay)	Tasa de fallos	AUC
red	Selección Miner	3	0.01	0.371	0.654
red2	Importancia	5	0.01	0.38	0.637
red3	Importancia	3	0.001	0.38	0.636
red4	Aleatoria 1	3	0.1	0.369	0.646
red5	Aleatoria 2	3	0.1	0.371	0.646
red6	Aleatoria 2	7	0.1	0.373	0.643
red7	Mejor con 10	3	0.001	0.370	0.654

Al comparar los resultados obtenidos en las Figuras 23 y 24. La red4 y red7 son las que presentan menor tasa de fallos, sin embargo al observar el área bajo la curva ROC la red4 tiene un valor más bajo que la red7. Por esta razón se selecciona como el mejor modelo la red7, que se configura con las variables del conjunto "mejor con 10", tiene 3 nodos y tasa de aprendizaje 0.001.

Con el fin de validar si se puede mejorar el resultado de la red, se analiza si se requiere parada anticipada de las iteraciones (early stopping) utilizando la macro red neuronal binaria. Una de las salidas de esta macro es un gráfico el cuál se encuentra en el Anexo II.

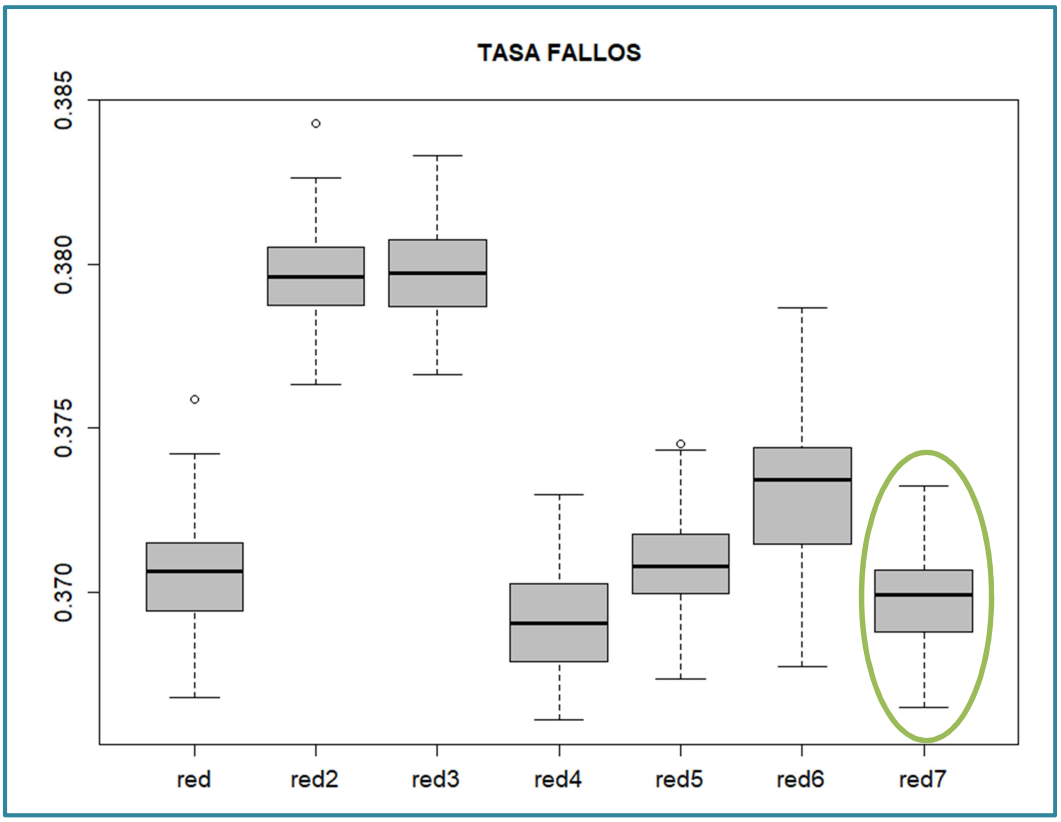


Figura 23. Resultados redes en R – tasa de fallos parte 1.

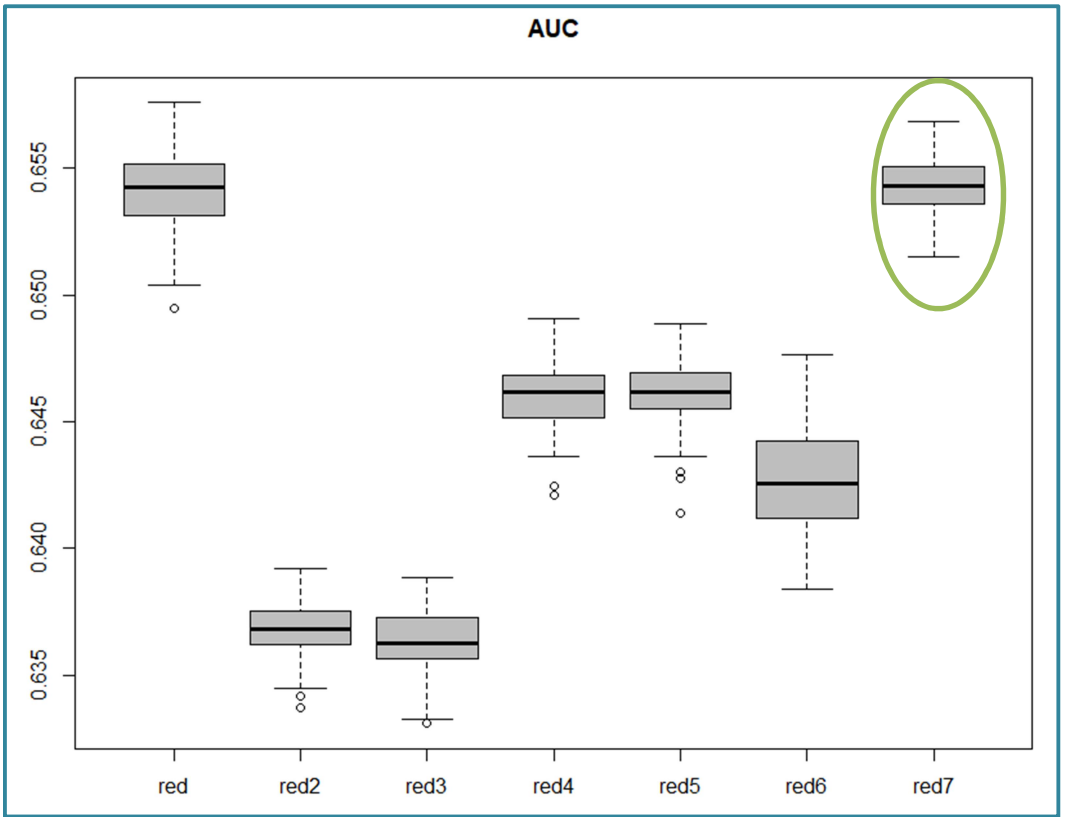


Figura 24. Resultados redes en R – AUC parte 1.

Adicionalmente se realizan pruebas cambiando el algoritmo de optimización y la función de activación para esto se ejecuta en SAS Base la macro Cruzada binaria

neural. En la Tabla 17, se relacionan las pruebas realizadas, el detalle de cada resultado se puede ver en el Anexo IV.

Tabla 17. Configuración de las redes en SAS Base parte 1.

Conjunto de variables	# Modelo	# Nodos	Algoritmo	Función de activación	momentum	Tasa de aprendizaje	Iteraciones decididas por parada anticipada
Selección Miner	Final8	3	Bprop	tanh	0.2	0.01	sin
Selección Miner	Final9	3	levmar	tanh	NA	NA	8
Selección Miner	Final10	3	levmar	tanh	NA	NA	sin
Selección Miner	Final11	3	levmar	log	NA	NA	sin
Selección Miner	Final12	3	levmar	log	NA	NA	8
Importancia	Final13	3	Bprop	tanh	0.2	0.01	sin
Importancia	Final14	5	Bprop	tanh	0.2	0.01	sin
Importancia	Final15	3	levmar	tanh	NA	NA	sin
Importancia	Final16	5	levmar	tanh	NA	NA	sin
Importancia	Final17	3	levmar	log	NA	NA	sin
Importancia	Final18	5	levmar	log	NA	NA	sin
Importancia	Final19	3	levmar	log	NA	NA	10
Aleatoria 1	Final20	3	Bprop	tanh	0.2	0.1	sin
Aleatoria 1	Final21	3	levmar	tanh	NA	NA	sin
Aleatoria 1	Final22	3	levmar	log	NA	NA	sin
Aleatoria 2	Final23	3	Bprop	tanh	0.2	0.1	sin
Aleatoria 2	Final24	3	levmar	tanh	NA	NA	11
Aleatoria 2	Final 25	7	levmar	tanh	NA	NA	10
Aleatoria 2	Final26	3	levmar	log	NA	NA	11
Mejor con 10	Final27	3	Bprop	tanh	0.2	0.001	8
Mejor con 10	Final28	3	levmar	tanh	NA	NA	8
Mejor con 10	Final29	3	levmar	log	NA	NA	8

Después de realizar las pruebas modificando la configuración de la red, no se logra mejorar el resultado obtenido, en la Figura 25 se muestra el mejor modelo de cada conjunto probado, siendo el modelo 12 el que menor sesgo presenta con un valor superior al 37% mientras en R se consigue un valor inferior al 37%.

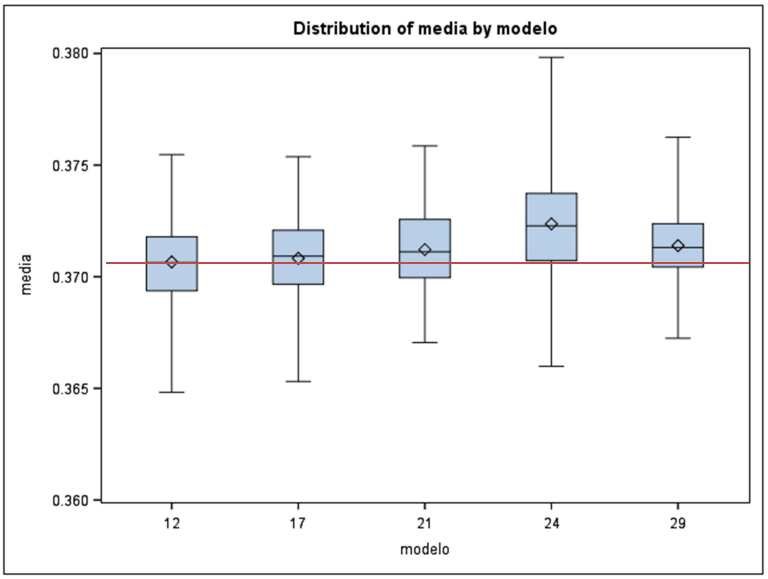


Figura 25. Resultados redes con SAS Base parte 1.

4.5.3 Bagging y Random Forest

Se utilizarán todos los conjuntos de variables probando para Bagging el número de iteraciones y tamaño de la muestra con todas las variables del conjunto y para

Random Forest, dado que la técnica se caracteriza por sortear aleatoriamente el número de variables a utilizar para la configuración de los árboles, se probará con N-1 variables hasta mínimo cuatro. Se utiliza la librería RandomForest de R y la función cruzadarfbin proporcionada por Portela, (2019).

Inicialmente, se analiza si se requiere parada anticipada que definirá el número de iteraciones para cada conjunto de variables, se prueban diferentes valores de tamaño de la muestra (sampsize): 2000, 3000 y 4000, y se varía el número de observaciones mínimas por nodo para el tamaño de muestra que presente mejor resultado. En la Tabla 18 se relacionan todas las configuraciones probadas y sus resultados. Para mayor detalle del análisis de parada anticipada, ver el Anexo V.

Tabla 18. Configuraciones bagging parte 1.

Modelo	Conjunto de variables	# Variables	# iteraciones	observaciones minimas por nodo	Tamaño de la muestra	Tasa de fallos	AUC
Bag	Miner	14	5500	30	2000	0.375	0.645
Bag2	Miner	14	5500	30	3000	0.377	0.641
Bag3	Miner	14	5500	30	4000	0.378	0.638
Bag4	Importancia	8	500	30	2000	0.387	0.625
Bag5	Importancia	8	500	30	3000	0.387	0.624
Bag6	Importancia	8	500	30	4000	0.387	0.623
Bag7	Aleatoria 1	15	500	30	2000	0.371	0.64
Bag8	Aleatoria 1	15	500	30	3000	0.374	0.64
Bag9	Aleatoria 1	15	500	30	4000	0.374	0.635
Bag10	Aleatoria 2	14	1000	30	2000	0.371	0.641
Bag11	Aleatoria 2	14	1000	30	3000	0.372	0.639
Bag12	Aleatoria 2	14	1000	30	4000	0.373	0.636
Bag13	Mejor con 10	10	6000	30	2000	0.376	0.642
Bag14	Mejor con 10	10	6000	30	3000	0.379	0.637
Bag15	Mejor con 10	10	6000	30	4000	0.375	0.634
Bag16	Miner	14	5500	20	2000	0.376	0.642
Bag17	Importancia	8	500	20	3000	0.387	0.623
Bag18	Aleatoria 1	15	500	20	2000	0.373	0.638
Bag19	Aleatoria 2	14	1000	20	2000	0.372	0.639
Bag20	Mejor con 10	10	6000	20	2000	0.372	0.64

Analizando los resultados que se muestran en las Figuras 26 y 27, el modelo Bag10 es el que presenta mejor tasa de fallos y la segunda mejor área bajo la curva ROC, se seleccionará como mejor modelo de Bagging.

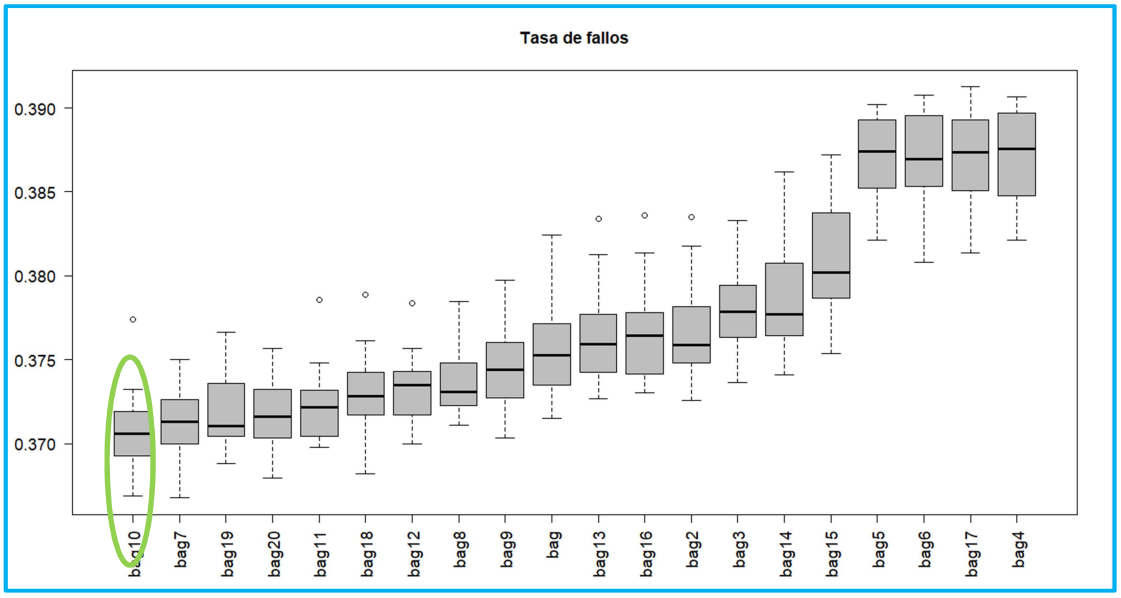


Figura 26. Resultados Bagging tasa de fallos parte 1.

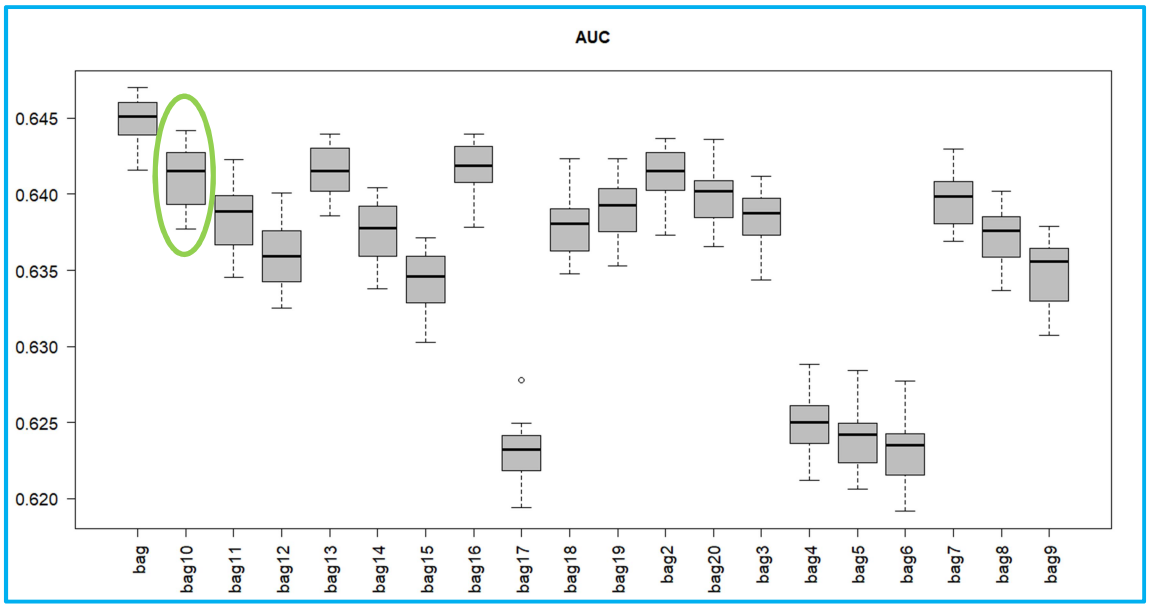


Figura 27. Resultados Bagging AUC parte 1.

Dado que random forest es una generalización del bagging, se utilizarán los parámetros tuneados de observaciones mínimas por nodo, tamaño de la muestra, y número de iteraciones obtenidos por Caret y se modificará el número de variables a sortear que se utiliza en el modelo. En la Tabla 19, se relacionan las configuraciones probadas para random forest.

Tabla 19. Configuraciones random forest parte 1.

Modelo	Conjunto de variables	# Variables	# iteraciones	observ. Min. por nodo	Tamaño de la muestra	Tasa de fallos	AUC
RF1	Miner	12	5500	30	2000	0.374	0.645
RF2	Miner	10	5500	30	2000	0.374	0.646
RF3	Miner	8	5500	30	2000	0.373	0.647
RF4	Miner	6	5500	30	2000	0.373	0.647
RF5	Miner	4	5500	30	2000	0.372	0.647
RF6	Importancia	7	500	30	3000	0.386	0.625
RF7	Importancia	6	500	30	3000	0.385	0.627
RF8	Importancia	5	500	30	3000	0.384	0.628
RF9	Importancia	4	500	30	3000	0.385	0.63
RF10	Aleatoria 1	13	500	30	2000	0.372	0.64
RF11	Aleatoria 1	11	500	30	2000	0.372	0.64
RF12	Aleatoria 1	9	500	30	2000	0.371	0.64
RF13	Aleatoria 1	7	500	30	2000	0.371	0.64
RF14	Aleatoria 1	5	500	30	2000	0.37	0.639
RF15	Aleatoria 2	12	1000	30	2000	0.37	0.641
RF16	Aleatoria 2	10	1000	30	2000	0.37	0.641
RF17	Aleatoria 2	8	1000	30	2000	0.37	0.641
RF18	Aleatoria 2	6	1000	30	2000	0.37	0.64
RF19	Aleatoria 2	4	1000	30	2000	0.371	0.639
RF20	Mejor con 10	8	6000	30	2000	0.376	0.643
RF21	Mejor con 10	6	6000	30	2000	0.374	0.644
RF22	Mejor con 10	4	6000	30	2000	0.372	0.645

Como se puede observar en las Figuras 28 y 29, el modelo que consigue la mayor área bajo la curva es el RF5 aunque no tiene la mejor tasa de fallos la diferencia es mínima con respecto a los demás modelos y se utilizará como mejor random forest.

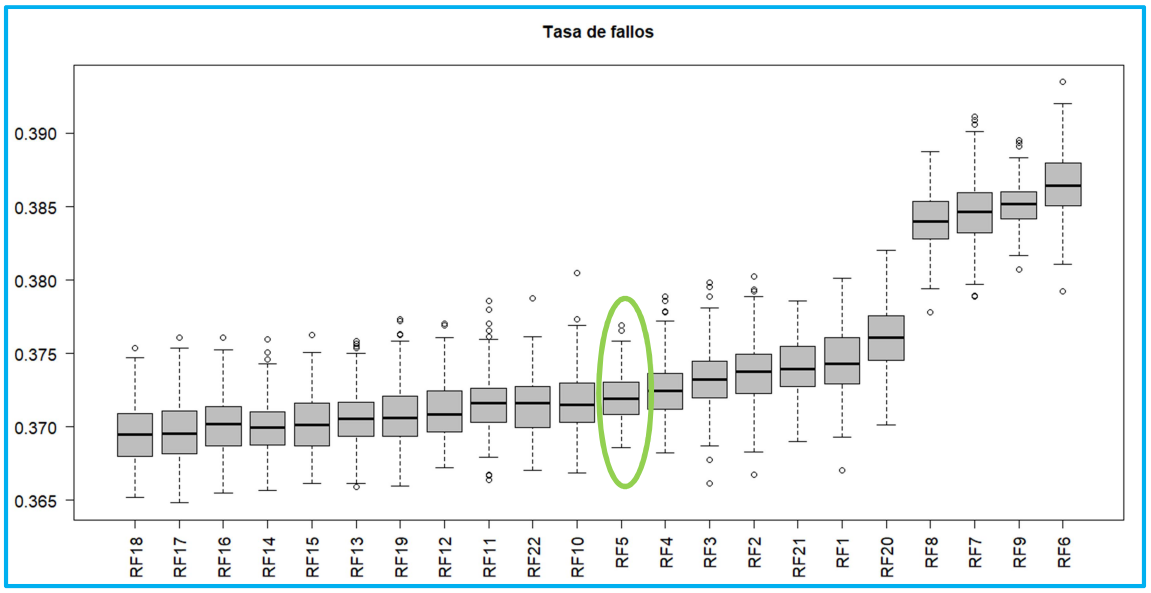


Figura 28. Resultados random forest tasa de fallos parte 1.

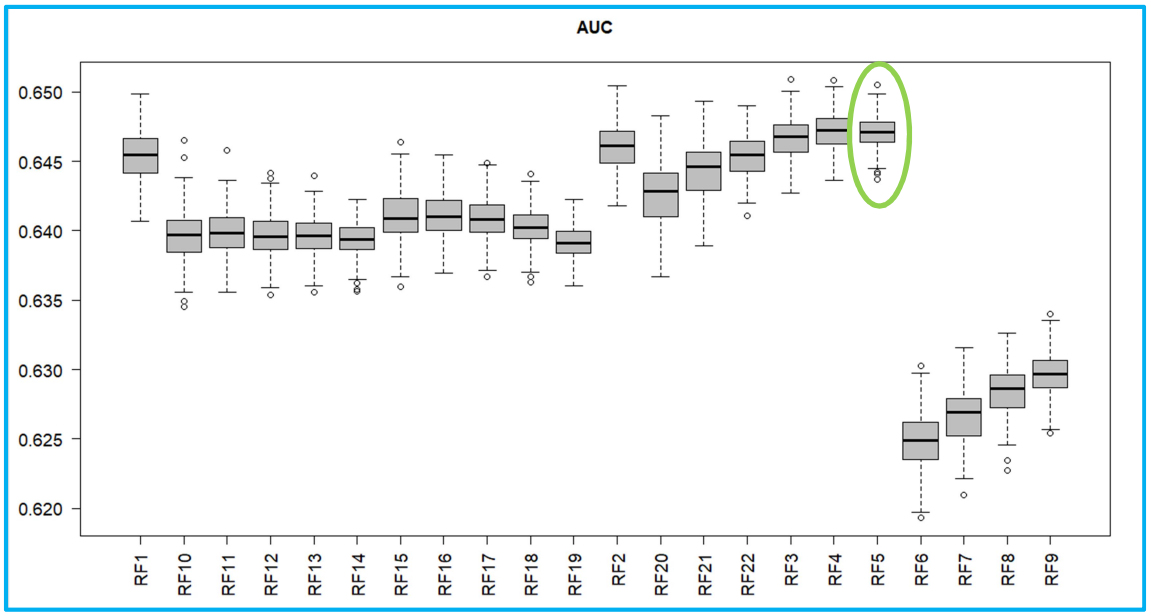


Figura 29. Resultados random forest AUC parte 1.

Dado que SAS base permite modificar las características de los árboles, se hacen pruebas con la configuración ganadora modificando la profundidad del árbol y el p-valor.

Tabla 20. Configuraciones bagging y random forest SAS parte 1.

Modelo	# Variables	# iteraciones	observaciones minimas por nodo	Tamaño de la muestra	Profundidad máxima	p valor
20	4	1000	30	40%	10	0.1
21	10	1000	30	40%	10	0.1
22	10	1000	30	70%	10	0.1
23	6	1000	30	40%	8	0.1
24	8	1000	30	40%	10	0.2

Como se puede observar en la Figura 30, al modificar los parámetros en SAS base, no se consigue obtener un mejor modelo con respecto al obtenido en el software R ya que la tasa de fallos es superior a 37% mientras que en la Figura 28 se puede ver que en R se consigue menor tasa de fallos.

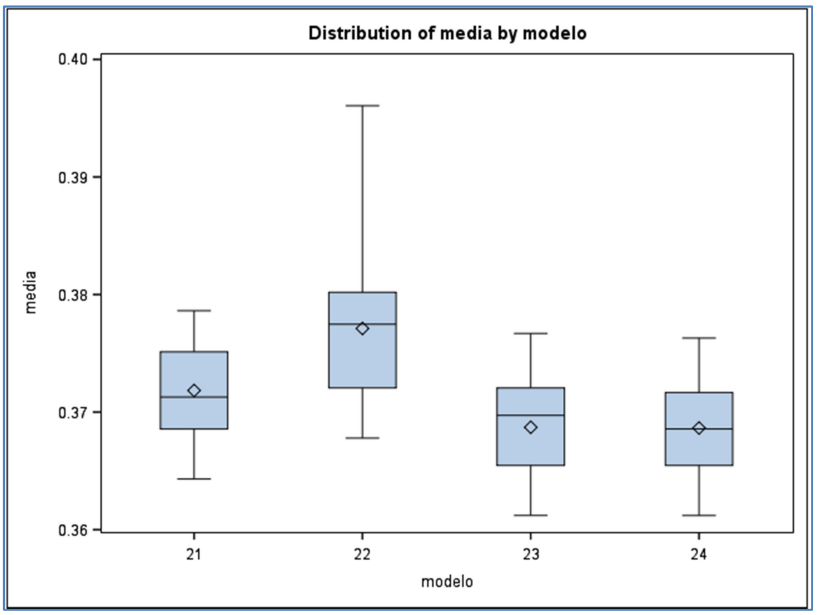


Figura 30. Resultados bagging y random forest SAS parte 1.

4.5.4 Incremento gradiente

Este algoritmo iterativo cuyo objetivo es minimizar el error, requiere que se defina la constante de regularización (Shrink), el número de iteraciones y la configuración del árbol. Para encontrar la configuración optima que logre mejoras en los resultados se deben realizar pruebas con diferentes valores, dependiendo de los resultados obtenidos, se van modificando parámetros para reducir sesgo o varianza, teniendo cuidado de no sobreajustar el modelo.

Se utiliza Caret de R para probar los diferentes valores de shrinkage (0.001, 0.01, 0.05, 0.1), mínimo de observaciones por nodo 10, 20 y 30, número de iteraciones 1000, 2000, y 3000. Los valores óptimos de estos parámetros, serán aquellos valores que logren maximizar la tasa de aciertos y serán utilizados en la configuración de los modelos a probar. También se realizan pruebas para verificar si se requiere parada anticipada (early stopping). Los resultados de estas pruebas se pueden ver en el Anexo VI.

Se utiliza validación cruzada repetida para probar las diferentes configuraciones e ir refinando el resultado, se realizan pruebas con la configuración obtenida con Caret y también con la constante de regularización baja (0.0001) e iteraciones altas (4000), se modifica la semilla de inicio de la validación cruzada para confirmar que no exista influencia del azar en los resultados, a medida que se van comprobando los resultados, se toma como base el mejor modelo y se ajustan parámetros como la constante de regularización y el número mínimo de observaciones por nodo. En la Tabla 21 se relacionan las configuraciones y sus resultados.

Tabla 21. Configuraciones incremento gradiente parte 1.

Modelo	Conjunto de variables	Cte de regularización	# iteraciones	observaciones minimas por nodo	Semilla	Tasa de fallos	AUC
GMB	Miner	0.01	2000	20	12345	0.369	0.656
GBM2	Miner	0.01	2000	20	12347	0.369	0.656
GMB3	Miner	0.0001	4000	20	12345	0.413	0.62
GMB4	Importancia	0.01	1000	10	12345	0.38	0.635
GMB5	Importancia	0.01	1000	10	12347	0.38	0.635
GMB6	Importancia	0.0001	4000	10	12345	0.413	0.62
GMB7	Aleatoria 1	0.01	2000	20	12345	0.368	0.649
GMB8	Aleatoria 1	0.01	2000	20	12347	0.368	0.649
GMB9	Aleatoria 1	0.0001	4000	20	12345	0.413	0.616
GMB10	Aleatoria 2	0.01	2000	30	12345	0.368	0.649
GMB11	Aleatoria 2	0.01	2000	30	12347	0.368	0.649
GMB12	Aleatoria 2	0.0001	4000	30	12345	0.413	0.616
GMB13	Mejor con 10	0.01	2000	10	12345	0.375	0.655
GMB14	Mejor con 10	0.01	2000	10	12347	0.37	0.655
GMB15	Mejor con 10	0.0001	4000	10	12345	0.413	0.62
GBM16	Aleatoria2	0.02	2000	30	12345	0.368	0.648
GBM17	Aleatoria2	0.015	2000	30	12345	0.368	0.649
GBM18	Aleatoria2	0.005	2000	30	12345	0.368	0.648

En cuanto a la tasa de fallos y valor del área bajo la curva ROC que se puede observar en las Figuras 31 y 32, se presenta un empate entre el modelo gbm10 y gbm11, se concluye que el modelo gbm10 por tener una variable menos y obtener el mismo resultado es el mejor modelo. Tiene una constante de regularización de 0.03, 1000 iteraciones y 20 0bservaciones mínimas por hoja.

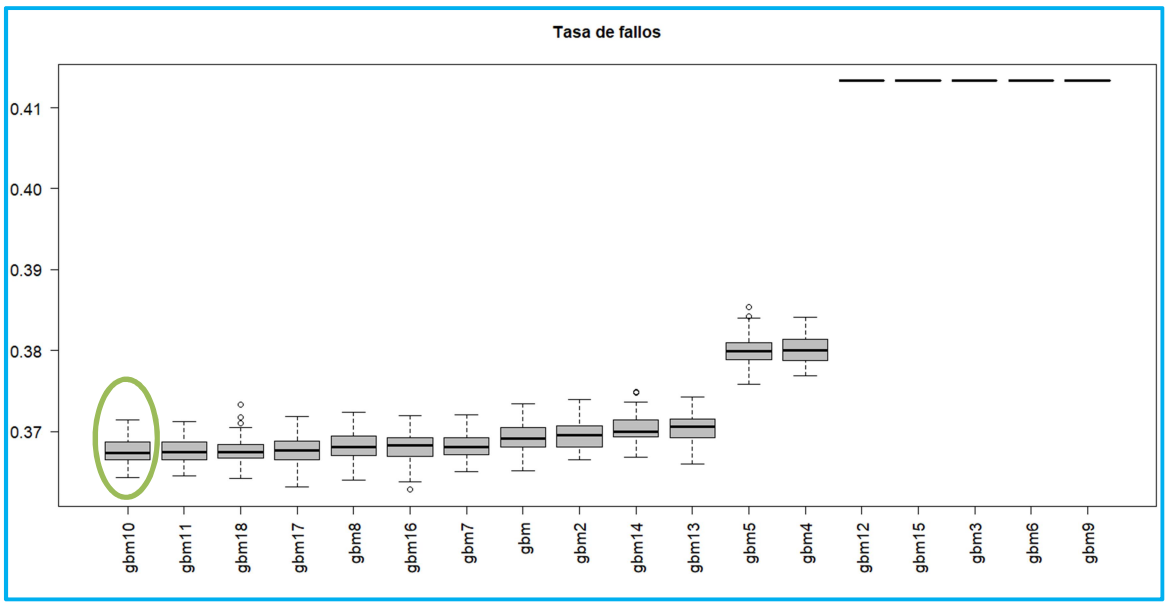


Figura 31. Resultados incremento gradiente tasa de fallos parte 1.

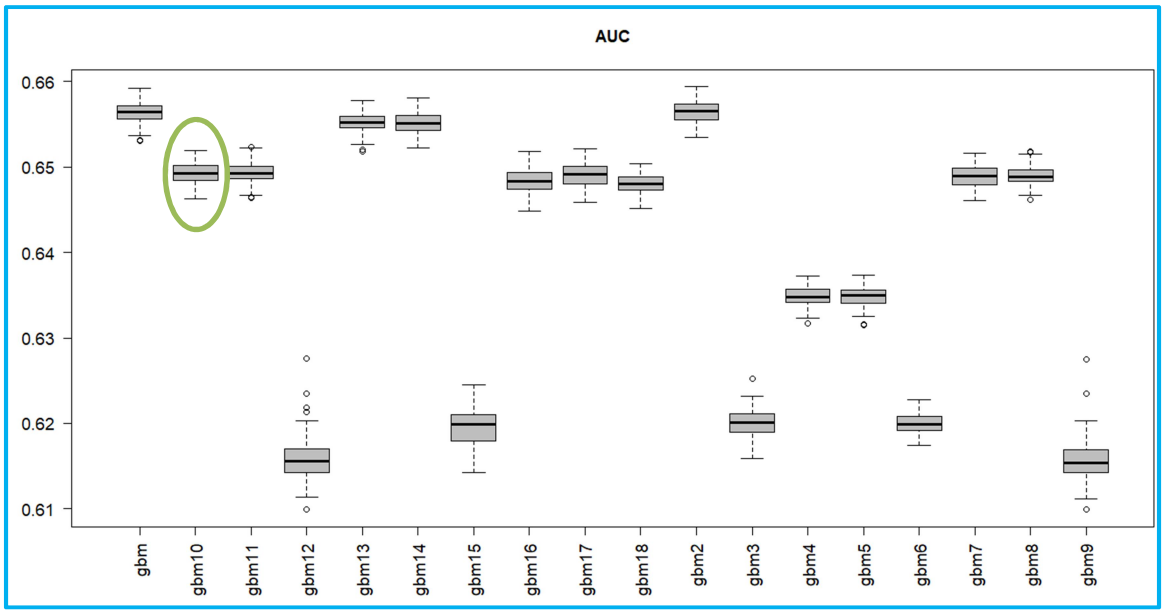


Figura 32. Resultados incremento gradiente AUC parte 1.

4.5.5 XGBoost

Como se comentó en la metodología, esta técnica es una modificación de incremento gradiente involucrando regularización o penalización en el momento de construir cada árbol, esta función de penalización previene el sobreajuste. Para determinar los valores de los parámetros se ejecuta Caret de R, probando las configuraciones sugeridas en Brownlee (2016) para cada conjunto de variables: constante de regularización: 0.01, 0.015, 0.025, 0.05 y 0.1, número de observaciones mínimas por nodo: 10, 20 y 30, numero de iteraciones: 1000, 2000 y 5000. Una vez se obtienen las configuraciones óptimas de estos parámetros se prueba modificando la máxima profundidad de los árboles: 5, 7, 9 y 12, la penalización gamma: 0,0.3, 0.5, 0.7 y 1, con sorteo de variables y observaciones de 0.8 y sin sorteo. En el Anexo VII se encuentra el detalle de los resultados para cada conjunto de variables. Una vez definidos los parámetros óptimos se realizan las pruebas.También se prueba modificando la semilla de la validación cruzada, el parámetro gamma, resultados se muestran en la Tabla 22.

Tabla 22. Configuraciones xgboost parte 1.

Modelo	Conjunto de variables	Cte de regularización	# iteraciones	observaciones minimas por nodo	Semilla	gamma	sorteo de variables	lamba	sorteo de observaciones	profundidad máxima	Tasa de fallos	AUC
XGBM1	Miner	0.01	1000	10	1234	0	no		no	5	0.374	0.651
XGBM2	Importancia	0.01	2000	20	12345	0	no		no	5	0.385	0.629
XGBM3	Aleatoria1	0.015	1000	30	12345	0	no		no	5	0.371	0.642
XGBM4	Aleatoria2	0.015	1000	20	12345	0	no		no	5	0.37	0.646
XGBM5	Mejor con 10	0.01	1000	20	12345	0	no		no	5	0.374	0.652
XGBM6	Miner	0.01	1000	10	12346	0	no		no	5	0.374	0.651
XGBM7	Importancia	0.01	2000	20	12347	0	no		no	5	0.385	0.629
XGBM8	Aleatoria1	0.015	1000	30	12347	0	no		no	5	0.372	0.642
XGBM9	Aleatoria2	0.015	1000	20	12347	0	no		no	5	0.371	0.646
XGBM10	Mejor con 10	0.01	1000	20	12347	0	no		no	5	0.37	0.647
XGBM11	Mejor con 10	0.01	1000	20	12347	0	0.8		no	5	0.374	0.651
XGBM12	Mejor con 10	0.01	1000	20	12345	0	0.8	10	no	5	0.373	0.653
XGBM13	Aleatoria2	0.01	1000	20	12345	0	no	10	no	5	0.37	0.647
XGBM14	Aleatoria2	0.01	1000	20	12345	0	0.8		no	5	0.369	0.647
XGBM15	Aleatoria2	0.01	1000	20	12345	0	no	10	no	5	0.369	0.647
XGBM16	Aleatoria2	0.01	1000	20	12345	0	0.8	10	no	5	0.369	0.648
XGBM16	Aleatoria2	0.01	1000	20	12345	0	0.8	10	no	10	0.369	0.648

Analizando los resultados en las Figuras 33 y 34, el modelo que mejores resultados presenta en cuanto a tasa de fallos es el xgbm16, si bien en cuanto al área bajo la curva ROC no tiene el mejor resultado la diferencia es mínima con respecto a los demás modelos. El modelo seleccionado, tiene una constante de regularización de 0.01, número de iteraciones 1000, número de observaciones mínimas por nodo de 20, gamma igual a cero, con sorteo de variables del 80%, lamba de 10 y sin sorteo de observaciones.

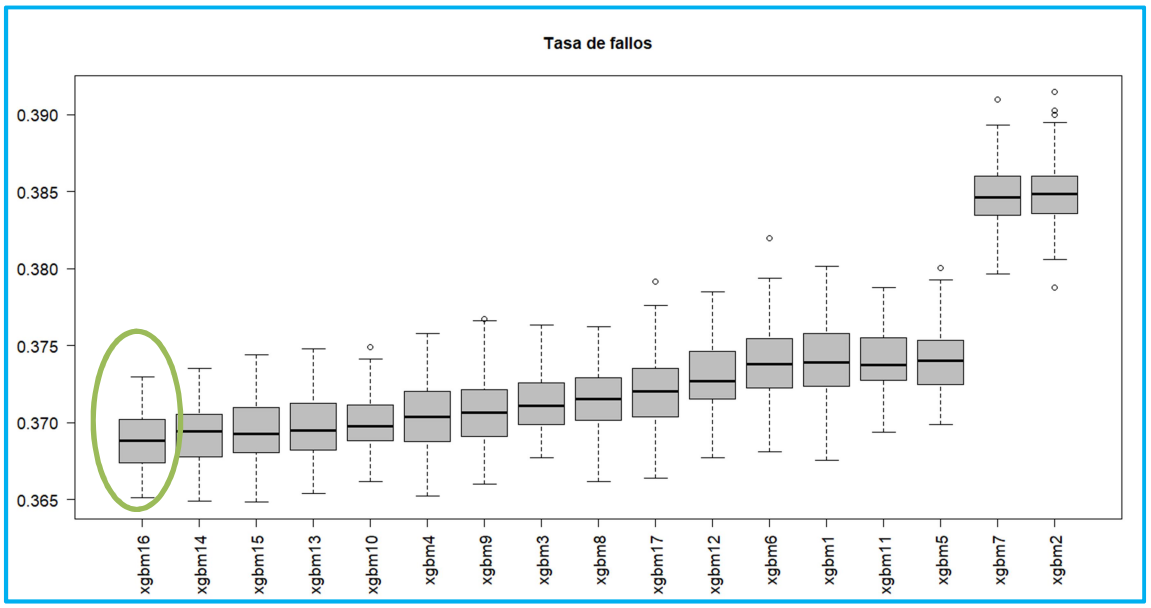


Figura 33. Resultados xgboost tasa de fallos parte 1.

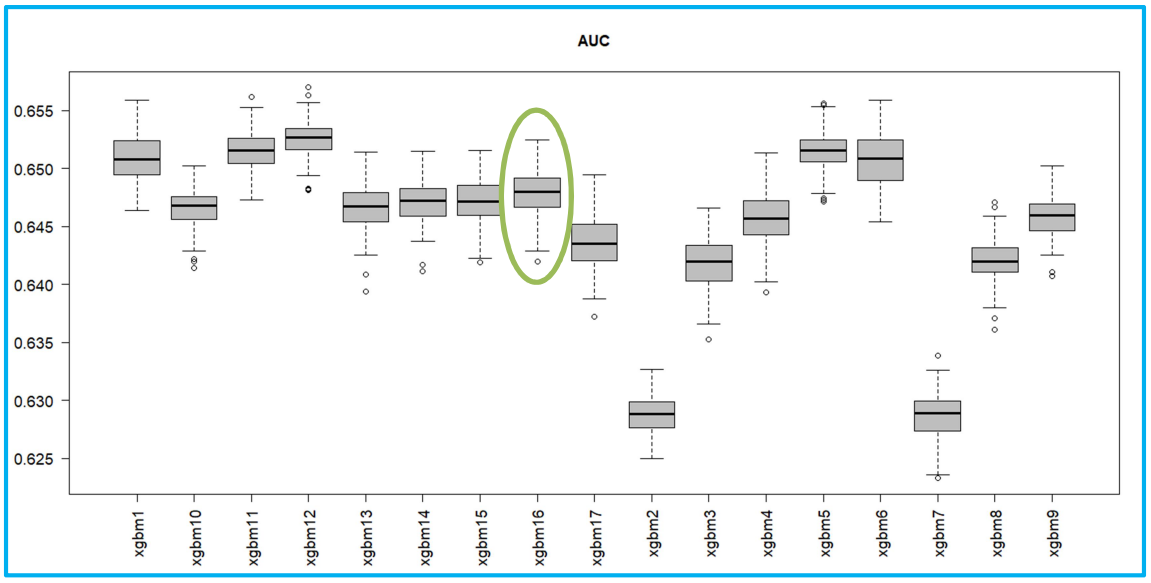


Figura 34. Resultados xgboost área bajo la curva ROC parte 1.

4.5.6 Máquinas de soporte vectorial (SVM)

Como se introdujo en la metodología, el objetivo de las máquinas de soporte vectorial es obtener el vector de parámetros W (que soportan la construcción de los hiperplanos), maximizando la distancia entre los dos hiperplanos de separación, este objetivo se logra a través de métodos clásicos de optimización permitiendo un

margen de error en la separación (ε) y un número máximo de observaciones que superen ese margen denominada constante de regularización (C). Se probará con SVM lineal, polinomial y RBF.

4.5.6.1 SVM Lineal

Para las máquinas de soporte vectorial lineales, se requiere definir la constante de regularización C, se utiliza de nuevo Caret de R para probar diferentes valores de este para cada conjunto de variables, como son: 0.001, 0.01, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 7, 10, 12, 20, 40, 60, 80 y 100. Se prueban por validación cruzada repetida todas las configuraciones obtenidas con Caret. Las configuraciones y resultados se muestran en la Tabla 23.

Tabla 23. Configuraciones probadas SVM Lineal parte 1.

Modelo	Conjunto de variables	Cte de regularización	Semilla	Tasa de fallos	AUC
SVML	Miner	0,05	1234	0.3869	0.5843
SVML2	Importancia	0,01	1234	0.38747	0.5691
SVML3	Aleatoria1	0,001	12345	0.3869	0.5922
SVML4	Aleatoria2	0,01	12345	0.3868	0.5862
SVML5	Mejor con 10	5	12345	0.3811	0.6314

En la Figura 35, se puede ver que el SVML5 logra una menor tasa de fallos y mayor área bajo la curva ROC. Con constante de regularización igual a cinco.

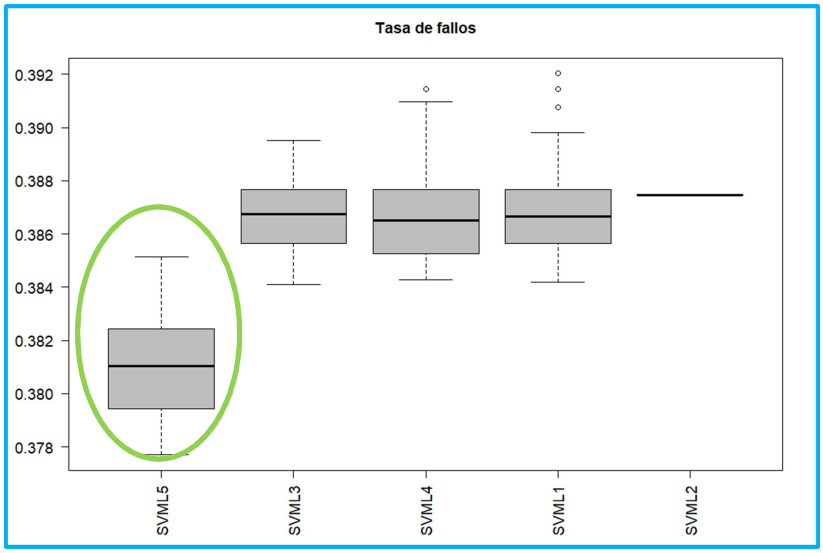


Figura 35. Resultados tasa de fallos SVM lineal parte 1.

4.5.6.2 SVM Polinimial

El kernel polinomial además de la constante de regularización, necesita el grado del polinomio que se utiliza para cambiar la dimensión y la escala. Al igual que en punto anterior se utiliza Caret para monitorear el desempeño de los diferentes valores de la constante de regularización, definir el grado del polinomio y la escala que mejor tasa de aciertos obtenga. Para encontrar estos valores óptimos se prueban: C 0.001, 0.01, 0.02, 0.03, 0.04, 0.05, 0.1, 1, 5, 10, 20 y 40, grado dos y tres, escala= 0.1, 0.5, 1, 2 y 5.

Tabla 24. Configuraciones obtenidas para SVM Polinomial parte 1.

Modelo	Conjunto de variables	Cte de regularización	Grados	Escala	Semilla
SVMPoly	Miner	10	2	5	12345
SVMPoly2	Importancia	0,001	3	0,5	12345
SVMPoly3	Aleatoria1	0.1	2	5	12345
SVMPoly4	Aleatoria2	0.1	2	5	12345
SVMPoly5	Mejor con 10	0.1	3	5	12345

Se hacen pruebas con las configuraciones, al parecer el algoritmo no converge después de 72 horas de ejecución no genera resultados, por lo cual se aborta la ejecución y se descarta la utilización de SVM Polinomial.

4.5.6.3 SVM Radial (RBF)

El kernel RBF Gaussiano requiere que se defina el valor de la constante de regularización y sigma. Se utiliza Caret para probar: C=0.01, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10, 30 y sigma=0.01, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10 y 30. Con las configuraciones óptimas se prueban por validación cruzada repetida y se analizan los resultados que se pueden observar en la Tabla 25.

Tabla 25. Configuraciones probadas SVM Radial parte 1.

Modelo	Conjunto de variables	Cte de regularización	Sigma	Semilla	Tasa de fallos	AUC
SVMRBF1	Miner	0.5	0.1	1234	0.3772	0.6250
SVMRBF2	Importancia	0.1	5	1234	0.3790	0.6101
SVMRBF3	Aleatoria1	0.1	0.2	1234	0.3778	0.6206
SVMRBF4	Aleatoria2	0.5	0.2	1234	0.3780	0.6172
SVMRBF5	Mejor con 10	30	0.2	1234	0.3723	0.6254

Como se muestra en la Figura 36, en cuanto al kernel RBF, el modelo que más se adapta a los datos y logra una menor tasa de fallos es el SVMRBF5 configurado con constante de regularización 30 y sigma 0.2.

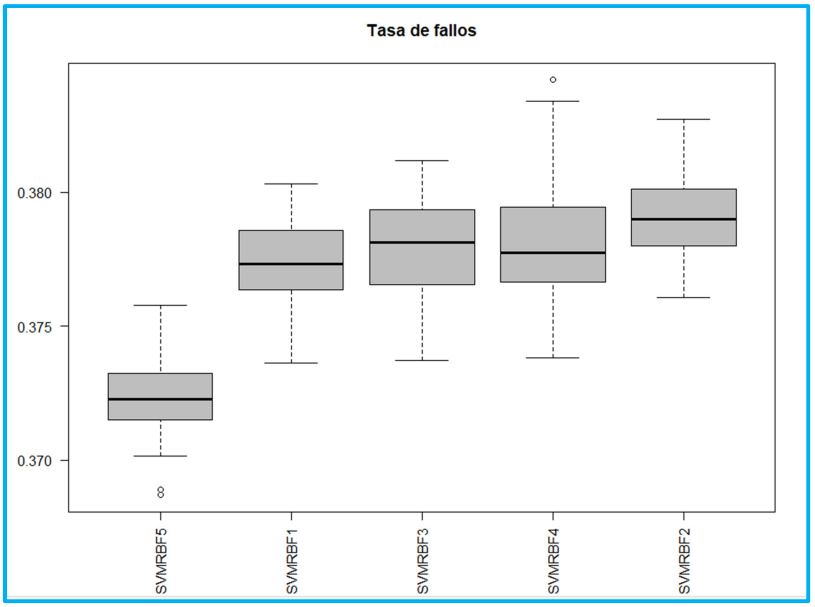


Figura 36. Resultados validación cruzada repetida SVM Radial parte 1.

4.5.7 Evaluación de los modelos

Una vez se tienen las configuraciones óptimas de cada algoritmo, se corren de nuevo con validación cruzada repetida, variando la semilla 100 veces y se comparan los resultados. En la Figura 37, se puede ver que la diferencia de la tasa de fallos entre los diferentes algoritmos no supera el 2%, el modelo obtenido con incremento gradiente es el que presenta menor tasa de fallos con una varianza inferior al 1%. En cuanto al área bajo la curva ROC que se observa en la Figura 38, existe una diferencia mínima entre la red que presenta mayor área e incremento gradiente. Se realizarán pruebas de ensamblado con estos modelos ganadores.

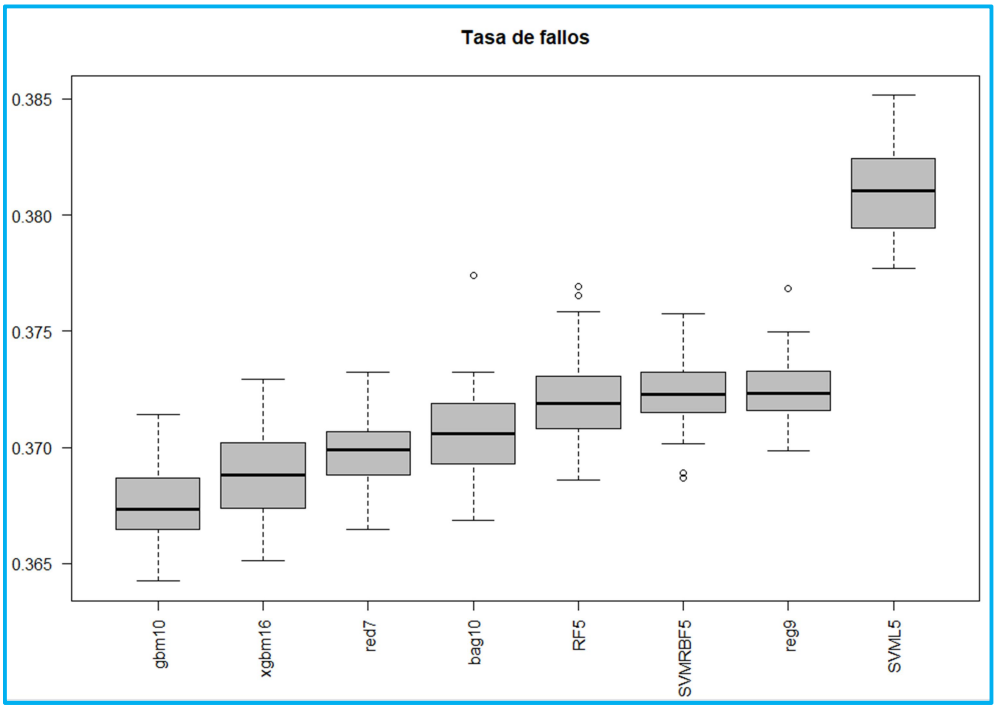


Figura 37. Comparación de los mejores modelos Tasa de fallos parte 1.

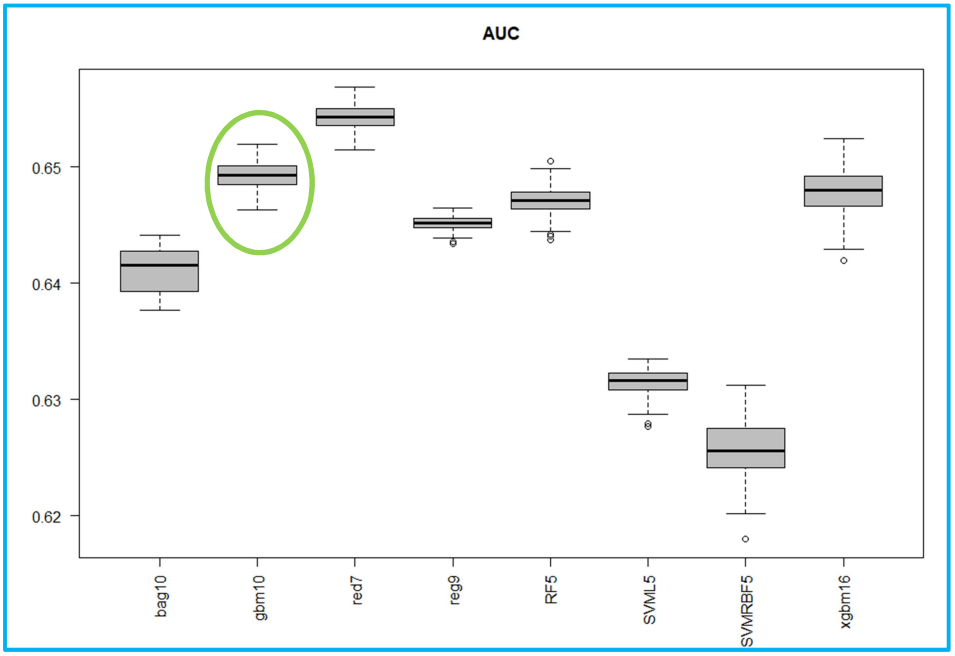


Figura 38. Comparación de los mejores modelos AUC parte 1.

4.5.8 Ensamblado

Su objetivo es construir predicciones a partir de la combinación de varios modelos, estos modelos a combinar serán los que mejor desempeño presentaron en esta parte del estudio, en la Tabla 26 se pueden ver las configuraciones probadas.

Tabla 26. Configuraciones ensamblado parte 1.

Modelo	Ensamble	Modelo	Ensamble
predi79	logi+avnnet	predi38	logi+rf+xgbm
predi80	logi+rf	predi39	logi+rf+svmLinear
predi81	logi+gbm	predi41	logi+rf+svmRadial
predi82	logi+xgbm	predi42	logi+gbm+xgbm
predi83	logi+svmLinear	predi43	logi+gbm+xgbm
predi85	logi+svmRadial	predi44	logi+gbm+svmLinear
predi86	avnnet+rf	predi46	logi+gbm+svmRadial
predi87	avnnet+gbm	predi47	logi+xgbm+svmLinear
predi18	avnnet+xgbm	predi49	logi+xgbm+svmRadial
predi19	avnnet+svmLinear	predi50	rf+gbm+svmLinear
predi21	avnnet+svmRadial	predi52	rf+gbm+svmRadial
predi22	rf+gbm	predi53	rf+xgbm+svmLinear
predi23	rf+xgbm	predi55	rf+xgbm+svmRadial
predi24	rf+svmLinear	predi56	rf+avnnet+gbm
predi26	rf+svmRadial	predi57	rf+avnnet+xgbm
predi27	gbm+xgbm	predi58	rf+avnnet+svmLinear
predi28	gbm+svmLinear	predi60	rf+avnnet+svmRadial
predi30	gbm+svmRadial	predi61	avnnet+gbm+svmLinear
predi31	logi+avnnet+rf	predi63	avnnet+gbm+svmRadial
predi32	logi+avnnet+gbm	predi64	logi+rf+gbm+avnnet
predi33	logi+avnnet+xgbm	predi65	logi+rf+xgbm+avnnet
predi34	logi+avnnet+svmLinear	predi66	logi+rf+xgbm+avnnet
predi36	logi+avnnet+svmRadial	predi67	logi+rf+xgbm+avnnet+svmLinear
predi37	logi+rf+gbm	predi69	logi+rf+xgbm+avnnet+svmRadial

Observando la Figura 40, el modelo que presenta mejores resultados se consigue ensamblando la red y xgboost, con un 36,5% de tasa de fallos. Estos dos modelos tienen una correlación cercana al 0.5 como se puede ver en la Figura 39. Esto indica que tienen diferencias y permiten aportar información que no se tenía incluida en los modelos individualmente. Cabe resaltar que la mejora obtenida con el ensamblado con respecto al peor modelo que sería la regresión logística es inferior al 1%, se concluye que la mejora en la tasa de fallos lograda con el ensamble no compensa la pérdida de interpretabilidad, por lo cual se selecciona como modelo ganador la regresión logística.

Con respecto al no modelo o no haber realizado ningún modelo para predecir, como se había comentado anteriormente la tasa de fallos sería del 43% (4447/10331). Esto significa que se logra mejorar este resultado en un 5,8% con la regresión logística. Si bien no es mucho, en términos monetarios puede representar una

cantidad considerable a tener en cuenta a la hora de hacer provisiones para pago a los prestadores de servicios de salud.

Se seleccionará este algoritmo para el cálculo de la primera parte del modelo final, que corresponde a la probabilidad de que un afiliado genere coste.

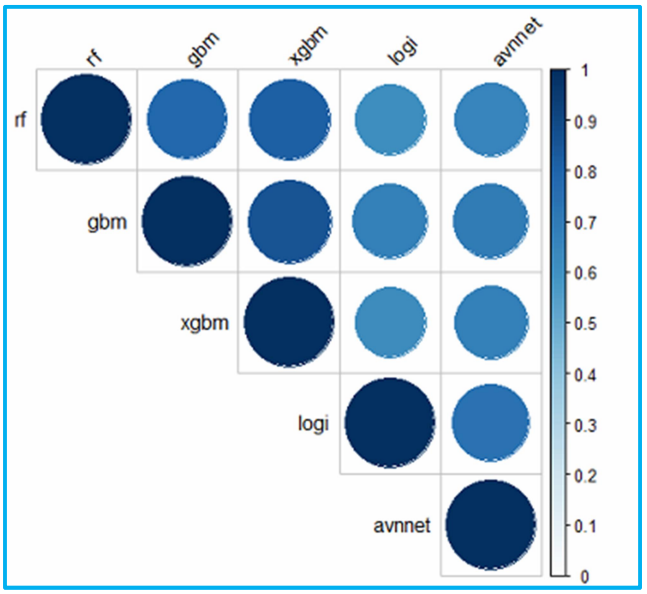


Figura 39. Correlación de modelos parte 1.

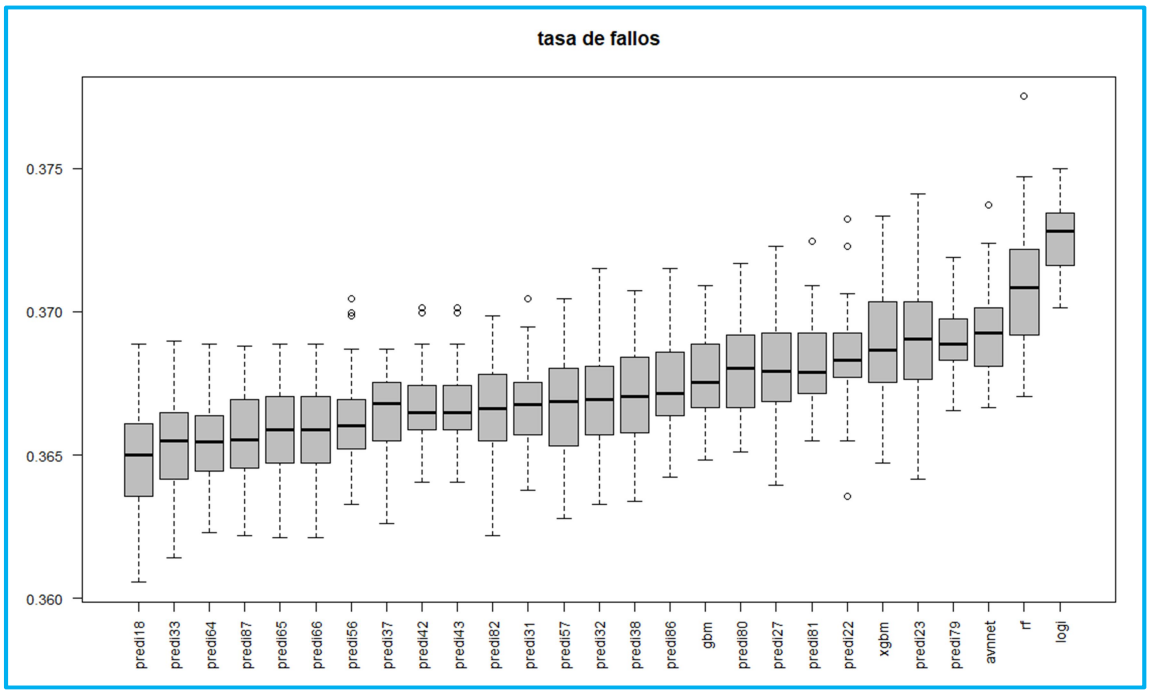


Figura 40. Resultados ensamblado con R parte 1.

4.6 Modelo de dos partes Variable objetivo continua (Parte 2).

4.6.1 Regresión

La segunda parte del modelo consiste en predecir el coste total anual, para este fin se utilizan los conjuntos de variables explicativas relacionados en la Tabla 27, la forma de selección ya fue explicada en el apartado correspondiente:

Tabla 27. Conjuntos de variables seleccionadas para la segunda parte.

Conjunto 1	Conjunto 2	Conjunto 3	Conjunto 4
Selección Miner	**Importancia de la variable**	**Selección aleatoria 1**	**Selección aleatoria 2**
edad2	edad	dias_afiliacion	dias_afiliacion
edad	edad2	edad2	edad2
edad_F	TI_G_enf_totales1	menos1	menos1
oncologia_adultos	TI_enf_totales1	dialisis	dialisis
TI_G_enf_totales1	TI_enf_totales2	oncologia_adultos	oncologia_adultos
TI_OPT_edad22	VIH	reumatologia_colagen	reumatologia_colagen
TI_OPT_edad4	dialisis	VIH	
TI_tipo2	oncologia_adultos	TI_enf_totales1	TI_enf_totales1
zona_2		TI_enf_totales2	TI_enf_totales2
dialisis		TI_OPT_edad3	TI_OPT_edad4

Teniendo las variables categóricas convertidas en dummies y las continuas estandarizadas, se utiliza la función **cruzadalin**, esta realiza validación cruzada repetida 200 veces para 10 grupos. En la Tabla 28, se muestran los resultados:

Tabla 28. Resultados y configuraciones regresión Lineal parte 2.

Modelo	Conjunto de variables	RMSE	R Cuadrado	#Variables
RL1	Selección miner	3044346	0.1823887	10
RL2	Importancia	3042589	0.1832355	8
RL3	Selección aleatoria 1	3044346	0.1823887	10
RL4	Selección aleatoria 2	3044346	0.1823887	9

En la Figura 41 se muestran los resultados de las pruebas en SAS Base, el modelo que presenta el menor error es el modelo cuatro obtenido con el conjunto de variables aleatoria 2. Como se puede observar en la Figura 42, los resultados para los modelos uno, tres y cuatro en R tienen el mismo valor. Se prefiere el conjunto de variables cuatro selección aleatoria 2 que tiene una variable menos que los otros dos conjuntos.

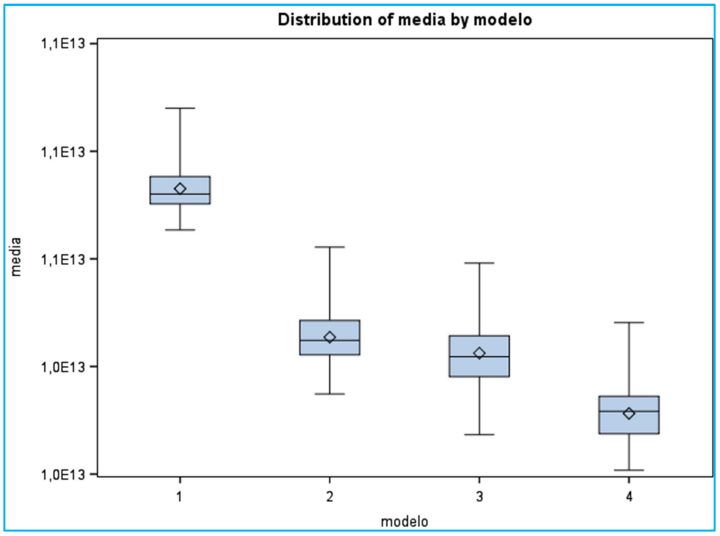


Figura 41. Resultados regresión con SAS Base parte 2.

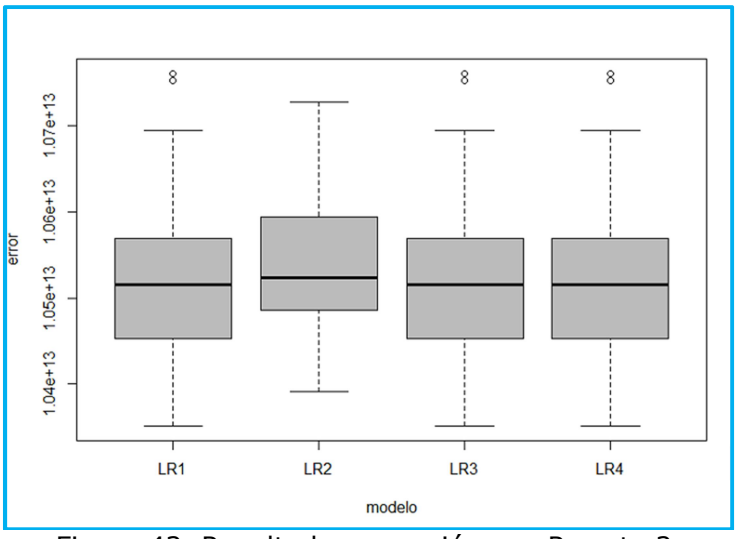


Figura 42. Resultados regresión con R parte 2.

Por principio de parsimonia se prefieren modelos más sencillos sobre los complejos, así es que se selecciona como mejor modelo el que menos variables tiene, que sería el conjunto cuatro: "Selección aleatoria 2". Este conjunto tiene nueve variables, dos continuas y siete categóricas.

4.6.2 Redes neuronales

Definición del número de nodos

Para definir el número de nodos a utilizar en la red, se calculan con la fórmula:

$$\#\frac{\min obs}{parametro} = h(k+1) + h + 1 \;\; donde\; h = \#nodos\; ocultos\; y\; k = \#\; variables\; independientes$$

Se tienen 4.447 observaciones, si se utiliza el 70% de los datos para entrenamiento se tendrían 3558 registros, aplicando la regla de 20 observaciones mínimas por nodo se tendría que el número mínimo de observaciones por parámetro seria = 3558/20 = 178. Se calcula el número de nodos máximo a utilizar para cada conjunto, en la Tabla 29 se muestran los resultados.

Tabla 29. Nodos calculados para cada conjunto de variables parte 2.

Conjunto de variables	# Variables continuas	# Variables categoricas	# Variables totales	# Nodos calculados
Selección Miner	3	7	10	15
Importancia de la variable	2	6	8	18
Aleatoria 1	2	8	10	15
Aleatoria 2	2	7	9	16

Para definir el número de nodos a utilizar en la red, se realizan pruebas con validación cruzada repetida con diferentes valores iniciando en 3 como mínimo, incrementando de a dos hasta llegar al máximo que coincide con el número de nodos calculados para conjunto. Utilizando la función Caret, que permite probar

diferentes valores para los nodos (size) y learning rate o decay de 0.01, 0.1 y 0.001. Se utiliza como máximo número de iteraciones de 200 y 10 repeticiones para cada combinación. También se prueba con SAS Base, el número de nodos con la macro variar, los resultados de las dos pruebas se pueden ver en la Tabla 30. Los gráficos obtenidos de las pruebas se pueden ver en el Anexo VIII.

Tabla 30. Número de nodos y tasa de aprendizaje a probar parte 2.

Conjunto de variables	Nombre	#Nodos R	Tasa de aprendizaje (decay)	#Nodos SAS
Conjunto 1	Selección Miner	15	0.1	15
Conjunto 2	Importancia	3	0.1	18
Conjunto 3	Aleatoria 1	3	0.1	15
Conjunto 4	Aleatoria 2	3	0.1	15

Una vez definido el número de nodos y la tasa de aprendizaje para cada conjunto de variables, se prueban en R las redes con validación cruzada repetida 200 veces utilizando la función cruzadaavnnetbin, las configuraciones probadas se muestran en la Tabla 31.

Tabla 31. Configuraciones redes neuronales parte 2.

Modelo	Conjunto de variables	# Nodos	Tasa de aprendizaje (decay)	ERROR	# Variables
red	Selección Miner	15	0.1	1,2169E+19	10
red2	Importancia	3	0.1	1,2169E+19	8
red3	Selección aleatoria 1	3	0.1	1,2169E+19	10
red4	Selección aleatoria 2	3	0.1	1,21694E+19	9

Como se muestra en la Figura 43, todos los modelos tienen el mismo resultado, por lo cual se selecciona como mejor modelo "mejor con 10", dado que tiene menos variables y se consigue el mismo resultado, este conjunto tiene dos variables continuas y siete categóricas. Se llega a la misma conclusión observando los resultados en SAS base de la Figura 44, donde los dos primeros modelos presentan una alta varianza y un sesgo similar.

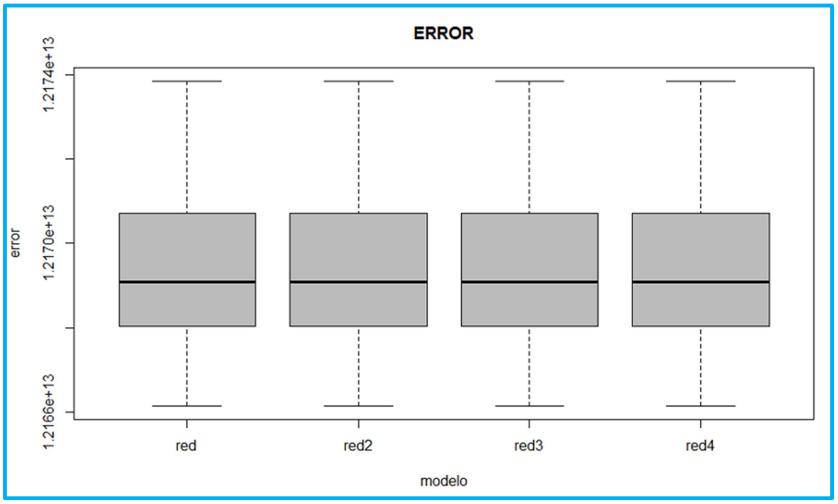


Figura 43. Resultados redes R parte 2.

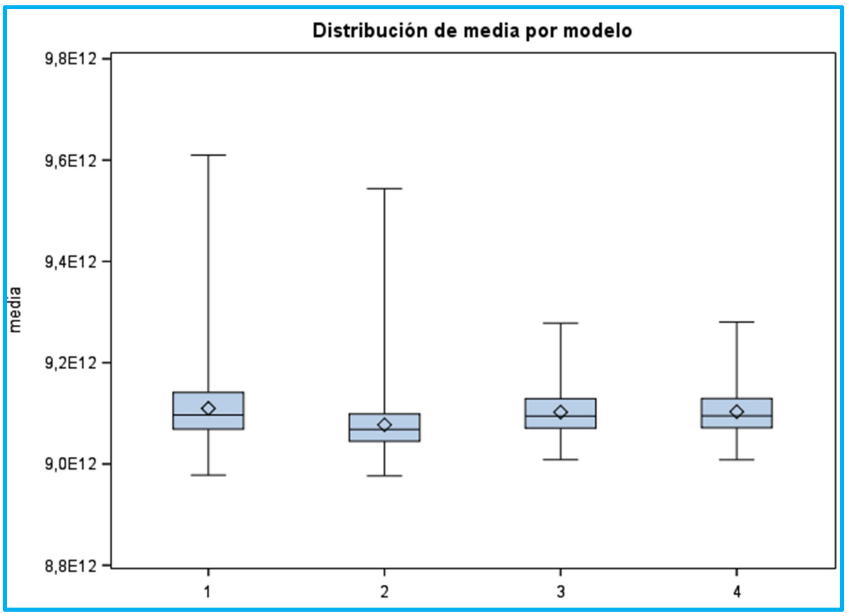


Figura 44. Resultados redes SAS parte 2.

4.6.3 Bagging y Random Forest

Se prueban todos los conjuntos de variables. Para Bagging se utiliza la totalidad de las variables del conjunto y para Random Forest, se probará con N-1 variables hasta mínimo cuatro. Se utiliza la librería RandomForest de R y la función cruzadarfbin proporcionada por Portela (2019).

Se prueban diferentes valores de tamaño de la muestra (sampsize): 1000, 2000 y 3000, observaciones mínimas por nodo y para definir el número de iteraciones se realiza análisis de parada anticipada que se puede ver en el Anexo X. La Tabla 32 muestra las configuraciones y resultados obtenidos.

Tabla 32. Configuraciones y resultados bagging parte 2.

Modelo	Conjunto de variables	# Variables	# iteraciones	observaciones minimas por nodo	Tamaño de la muestra	RMSE	R cuadrado
Bag	Selección Miner	10	400	30	1000	324800	0.1213702
Bag2	Selección Miner	10	400	30	2000	3285789	0.1107601
Bag3	Selección Miner	10	400	30	3000	3317489	0.1023968
Bag4	Importancia	8	500	30	1000	3220522	0.1329385
Bag5	Importancia	8	500	30	2000	3248782	0.12752
Bag6	Importancia	8	500	30	3000	3254963	0.126179
Bag7	Selección aleatoria 1	10	1000	30	1000	3199390	0.1406122
Bag8	Selección aleatoria 1	10	1000	30	2000	3213219	0.1404286
Bag9	Selección aleatoria 1	10	1000	30	3000	3221982	0.1401427
Bag10	Selección aleatoria 2	9	1000	30	1000	3222643	0.1300283
Bag11	Selección aleatoria 2	9	1000	30	2000	3230072	0.1328032
Bag12	Selección aleatoria 2	9	1000	30	3000	3237559	0.13222

De acuerdo con los resultados y observando la Figura 41, la configuración que logra un menor error es bag7, con 10 como número de observaciones mínimas por nodo, número de iteraciones 1000 y tamaño de muestra 1000. .

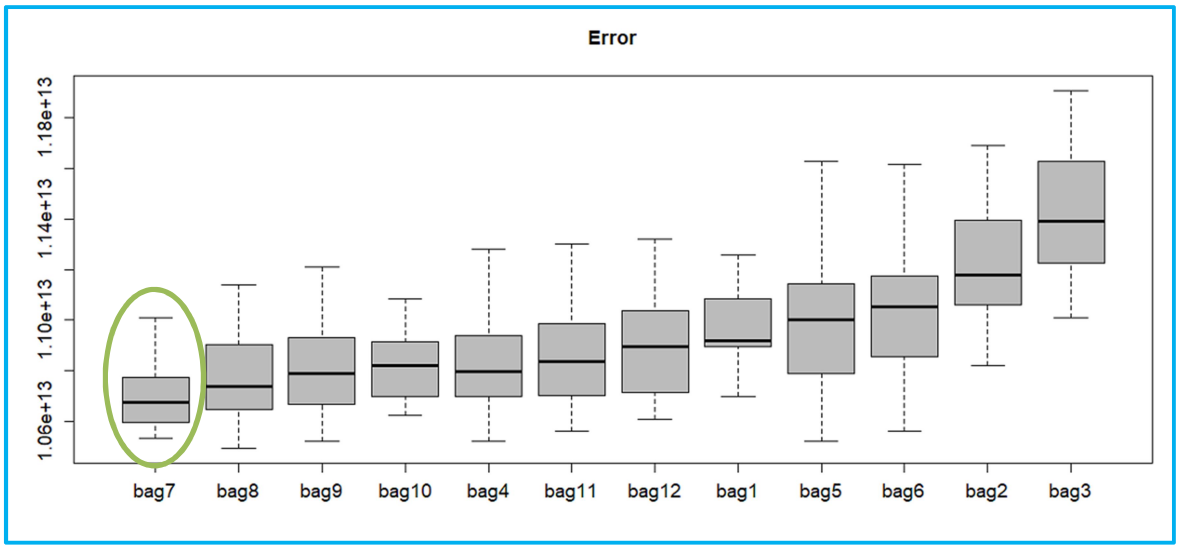


Figura 41. Resultados bagging parte 2.

En la Tabla 33 se relacionan las configuraciones probadas por validación cruzada repetida 200 veces con random forest, modificando los parámetros de observaciones mínimas por nodo, tamaño de la muestra, número de variables y número de iteraciones.

Tabla 33. Configuraciones random forest parte 2.

Modelo	Conjunto de variables	# Variables	# iteraciones	observ. Min. por nodo	Tamaño de la muestra	RMSE	R cuadrado
RF1	Selección Miner	8	400	30	1000	3239110	0.1231502
RF2	Selección Miner	6	400	30	1000	3225117	0.126532
RF3	Selección Miner	4	400	30	1000	3213371	0.1302656
RF4	Importancia	6	500	30	1000	3211320	0.1358513
RF5	Importancia	4	500	30	1000	3203157	0.1387904
RF6	Selección aleatoria 1	8	1000	30	1000	3192937	0.1428964
RF7	Selección aleatoria 1	6	1000	30	1000	3186240	0.146091
RF8	Selección aleatoria 1	4	1000	30	1000	3182514	0.148238
RF9	Selección aleatoria 2	8	1000	30	1000	3215857	0.1322293
RF10	Selección aleatoria 2	6	1000	30	1000	3204947	0.1368318
RF11	Selección aleatoria 2	4	1000	30	1000	3197198	0.1402365
RF12	Selección aleatoria 1	4	1000	20	1000	3176306	0.1526754

Como se puede observar en la Figura 42, se selecciona como mejor modelo el RF12, con cuatro variables, 1000 iteraciones, observaciones mínimas por nodo 20 y tamaño de la muestra 1000, utilizando el conjunto de datos selección aleatoria1.

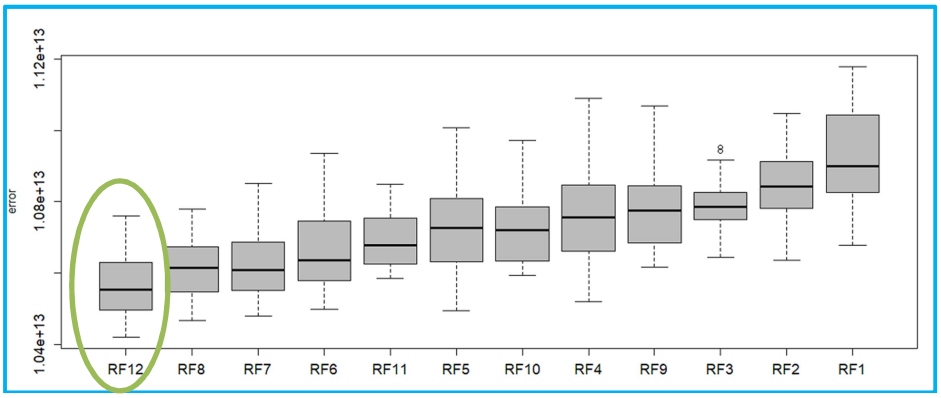


Figura 42. Resultados Random forest parte 2.

4.6.4 Incremento gradiente

Este algoritmo iterativo cuyo objetivo es minimizar el error, requiere que se defina la constante de regularización (Shrink), el número de iteraciones y la configuración del árbol. Se utiliza Caret de R, con validación cruzada repetida, para probar los diferentes valores: shrinkage=0.001, 0.01, 0.05, 0.1, mínimo de observaciones por nodo 10, 20 y 30, número de iteraciones 500,1000, 2000 y 3000. Los valores óptimos de estos parámetros, serán aquellos valores que logren minimizar el error cuadrático medio. También se realizan pruebas para verificar si se requiere parada anticipada (early stopping). Para ver las gráficas de los resultados de Caret y parada anticipada remitirse al Anexo XI. En la Tabla 34 se muestran las configuraciones probadas con sus resultados.

Tabla 34. Configuraciones incremento gradiente parte 2.

Modelo	Conjunto de variables	Cte de regularización	# iteraciones	observaciones minimas por nodo	Semilla	RMSE	R cuadrado
GBM1	Selección Miner	0.001	2000	20	12345	3240488	0.1139571
GBM2	Importancia	0.001	2000	10	12345	3244448	0.1139332
GBM3	Selección aleatoria 1	0.01	1000	30	12345	3244601	0.115501
GBM4	Selección aleatoria 2	0.01	1000	30	12345	3244464	0.1156084
GBM5	Selección aleatoria 2	0.01	1000	30	12347	3233623	0.1157467

Como se puede observar en la Figura 43, el modelo que tiene menor error cuadrático medio, es el GMB5. Se probó variando la semilla aleatoria de la validación cruzada y el resultado sigue siendo consistente, por lo que se selecciona como el mejor modelo.

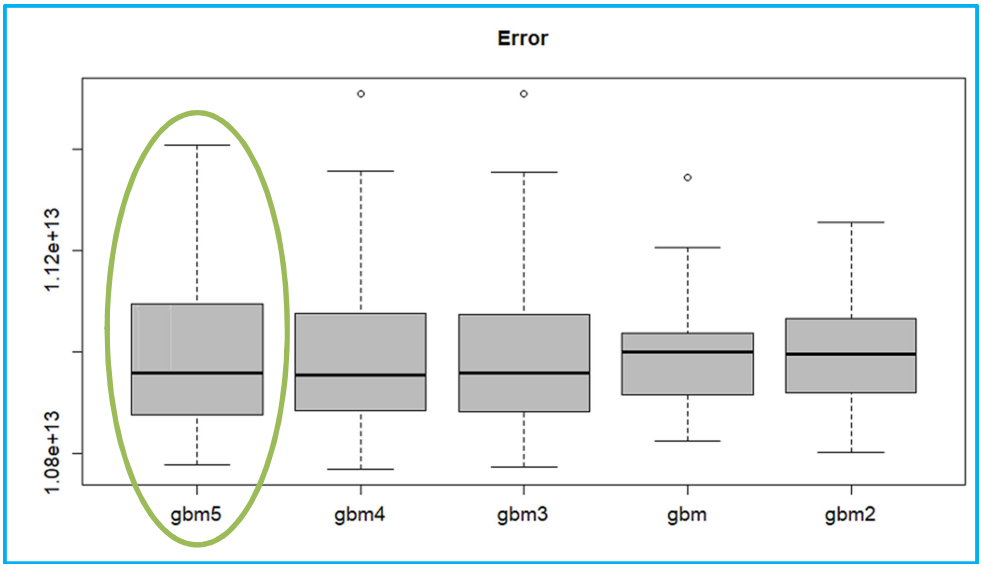


Figura 43. Resultados incremento gradiente parte 2.

4.6.5 XGBoost

Para determinar los valores de los parámetros se ejecuta Caret de R para cada conjunto de variables, probando las configuraciones sugeridas en (Brownlee, 2016) constante de regularización: 0.01, 0.015, 0.025, 0.05 y 0.1, número de observaciones mínimas por nodo: 5, 10, 15 , 20 y 30, número de iteraciones 500, 1000, 2000 y 3000, máxima profundidad de los árboles: 5, 7, 9 y 12, penalización

gamma 0, 0.3, 0.5, 0.7 y 1, con sorteo de variables y observaciones de 0.8 y sin sorteo de ambas. También se realizan otras validaciones como por ejemplo, si se requiere parada anticipada (early stopping) y análisis de valor óptimo para el parámetro lambda. Ver los resultados detallados en Anexo XII. Se prueba modificando la semilla de la validación cruzada, sorteo de variables y sorteo de observaciones.

En la Tabla 35, se relacionan las configuraciones probadas y resultados para xgboost, el mejor modelo se obtiene con el conjunto de variables aleatoria 2, modelo XGBM8. En la Figura 44 se puede observar que es el modelo que menor error presenta.

Tabla 35. Configuraciones y resultados xgboost parte 2.

Modelo	Conjunto de variables	Cte de regularización	# iteraciones	observaciones minimas por nodo	Semilla	gamma	sorteo de variables	lambda	sorteo de observaciones	prof. máxima	RMSE	R cuadrado
XGBM1	Miner	0.01	300	30	1234	1	no		no	5	3264457	0.1056
XGBM2	Importancia	0.01	300	30	1234	1	no		no	5	3273637	0.0997
XGBM3	Selección aleatoria 1	0.01	300	30	1234	1	no		no	5	3239768	0.1206
XGBM4	Selección aleatoria 2	0.01	300	30	1234	1	no		no	5	3238086	0.1216
XGBM5	Selección aleatoria 2	0.01	300	20	1234	1	no		no	5	3273998	0.1088
XGBM6	Selección aleatoria 2	0.01	300	30	1234	1	0.8		no	5	3233652	0.1237
XGBM7	Selección aleatoria 2	0.01	300	30	1234	1	0.8	10	no	5	3227543	0.1243
XGBM8	Selección aleatoria 2	0.01	300	30	1234	1	0.8	10	0.8	5	3217937	0.1286
XGBM9	Selección aleatoria 2	0.01	300	30	1234	1	0.8	10	0.8	10	3224172	0.1251
XGBM10	Importancia	0.01	300	30	1234	0.3	0.8		0.8	5	3262182	0.1042
XGBM11	Selección aleatoria 1	0.01	300	30	1234	0.3	no		0.8	5	3228251	0.1257
XGBM12	Selección aleatoria 2	0.01	300	20	1234	0.7	no		0.8	5	3228216	0.1261

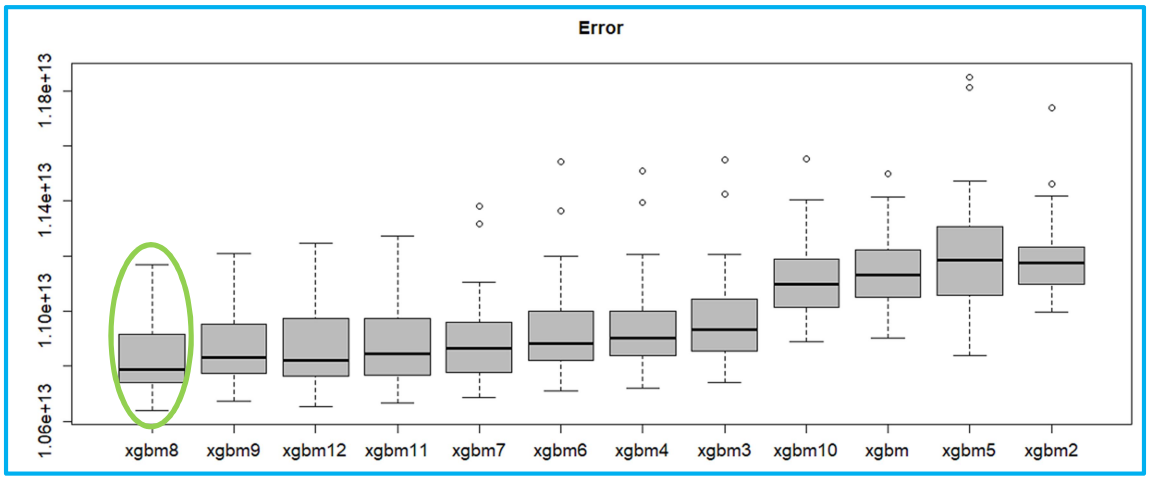


Figura 44. Resultados xgboost parte 2.

4.6.6 Máquinas de soporte vectorial (SVM)

4.6.6.1 SVM Lineal

Para las máquinas de soporte vectorial lineales, se requiere definir la constante de regularización C, se utiliza de nuevo Caret de R para probar diferentes valores de esta constante, como son: 0.001, 0.01, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 7, 10, 12, 20, 40, 60, 80 y 100 para cada conjunto de variables. Se prueban por validación cruzada repetida todas las configuraciones obtenidas con Caret. Las configuraciones y resultados se muestran en la Tabla 36.

Tabla 36. Configuraciones SVM Linea parte 2l.

Modelo	Conjunto de variables	Cte de regularización	Semilla	RMSE	R cuadrado
SVML	Miner	0.5	1234	3276367	0.1146259
SVML2	Importancia	0.01	1234	3215936	0.1472315
SVML3	Aleatoria1	2	1234	3216960	0.1465932
SVML4	Aleatoria2	2	1234	3178460	0.1659992

El modelo que logra una menor tasa de error es el SVML4, que utiliza el conjunto de variables aleatoria 2, constante de regularización 2, la comparación de los modelos probados se puede ver en la Figura 45.

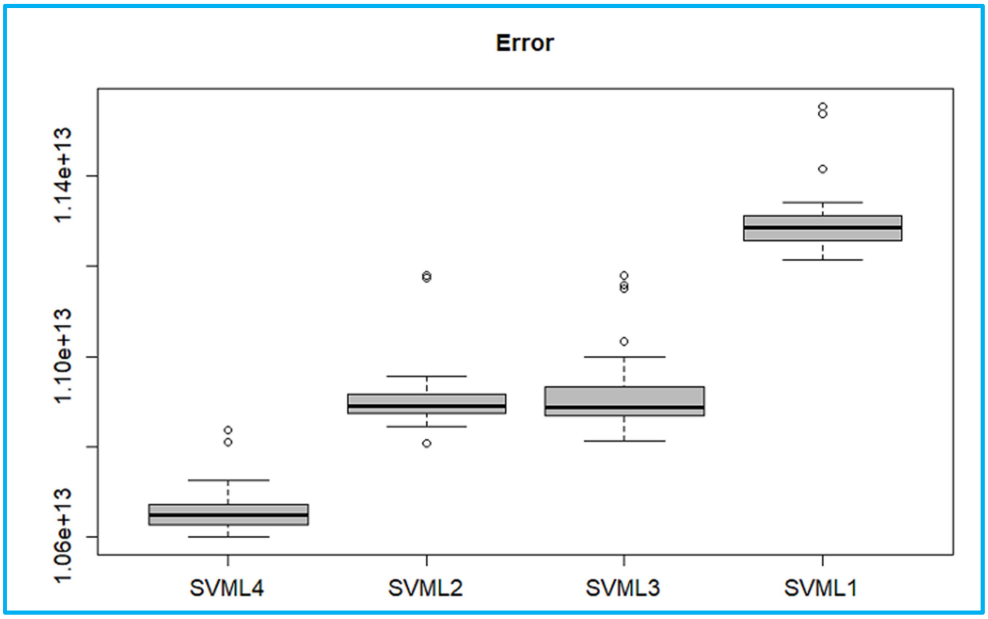


Figura 45. Resultados SVM Lineal parte 2.

4.6.6.2 SVM Radial (RBF)

El RBF requiere que se defina el valor de la constante de regularización y sigma, de nuevo se utiliza Caret para probar: C=0.01, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, y 10 y sigma= 0.01, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10 y 30. Se prueban por validación cruzada repetida 200 veces las configuraciones obtenidas con Caret que se relacionan en la Tabla 37.

Tabla 37. Configuraciones SVM Radial parte 2.

Modelo	Conjunto de variables	Cte de regularización	Sigma	Semilla	RMSE	R cuadrado
SVMRBF1	Miner	10	0.05	12345	3248922	0.1225919
SVMRBF2	Importancia	10	0.2	12345	3207364	0.1485349
SVMRBF3	Aleatoria1	10	0.1	12345	3206025	0.1495815
SVMRBF4	Aleatoria2			12345	3160860	0.1681768

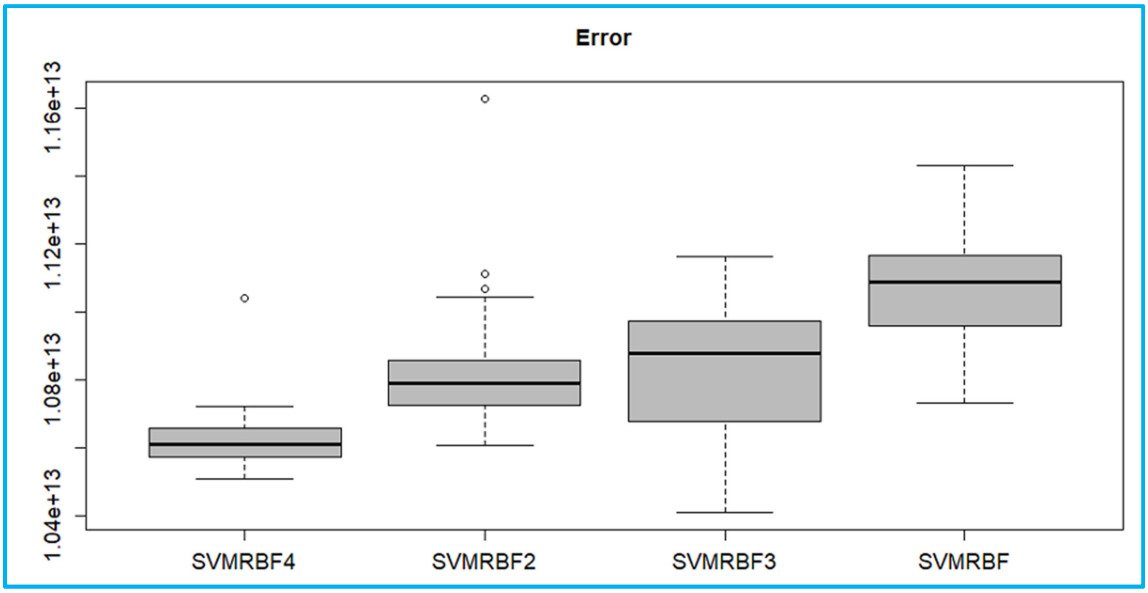


Figura 46. Resultados SVM Radial parte 2.

Como se muestra en la Figura 46 en cuanto al kernel RBF, el modelo que más se adapta a los datos y logra un menor error es el SMVRBF4 configurado con constante de regularización 10 y sigma 0.01.

4.6.7 Evaluación de los modelos

Una vez se tienen las configuraciones óptimas de cada algoritmo, se corren de nuevo con validación cruzada repetida, variando la semilla 100 veces y se comparan los resultados. Como se puede ver en la Figura 54, la regresión es la que tiene menor error cuadrático medio, la diferencia con los algoritmos basados en arboles no es muy significativa. En este caso la red no logra competir con los demás algoritmos. Se realizarán pruebas de ensamblado con estos modelos ganadores.

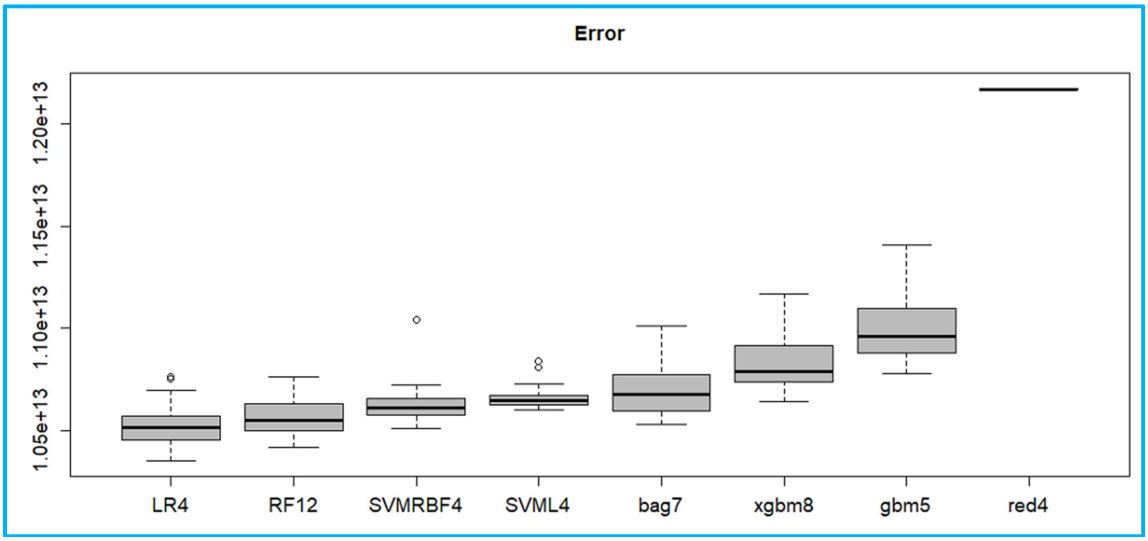


Figura 54. Comparación de los mejores modelos parte 2.

4.6.8 Ensamblado

En la Tabla 38 se listan las combinaciones de modelos utilizadas para ensamble.

Tabla 38. Configuración ensamblado parte 2.

Modelo	Ensamble	Modelo	Ensamble
predi81	reg+avnnet	predi99	reg+avnnet+svm
predi82	reg+rf	predi100	reg+rf+gbm
predi83	reg+gbm	predi101	reg+rf+xgbm
predi84	reg+xgbm	predi102	reg+rf+svm
predi85	reg+svm	predi103	rf+avnnet+gbm
predi86	avnnet+rf	predi104	rf+gbm+xgbm
predi87	avnnet+gbm	predi105	rf+gbm+xgbm
predi88	avnnet+xgbm	predi106	svm+gbm+xgbm
predi89	avnnet+svm	predi107	reg+gbm+xgbm
predi90	rf+gbm	predi108	reg+gbm+svm
predi91	rf+xgbm	predi109	reg+xgbm+svm
predi92	rf+svm	predi110	reg+rf+gbm+avnnet
predi93	gbm+xgbm	predi111	reg+rf+gbm+xgbm
predi94	gbm+svm	predi112	reg+svm+gbm+xgbm
predi95	xgbm+svm	predi113	reg+avnnet+gbm+xgbm
predi96	reg+avnnet+rf	predi114	reg+avnnet+svm+xgbm
predi97	reg+avnnet+gbm	predi115	reg+rf+gbm+xgbm
predi98	reg+avnnet+xgbm		

En la Figura 55 se muestran los resultados del ensamblado, el modelo predi109 de ensamble que incluye regresión, xgboost y máquinas de soporte vectorial, es el que logra el mejor resultado. Cabe destacar que de nuevo la diferencia con respecto a la regresión es mínima y dado que se sacrifica interpretabilidad del modelo se selecciona la regresión como modelo ganador.

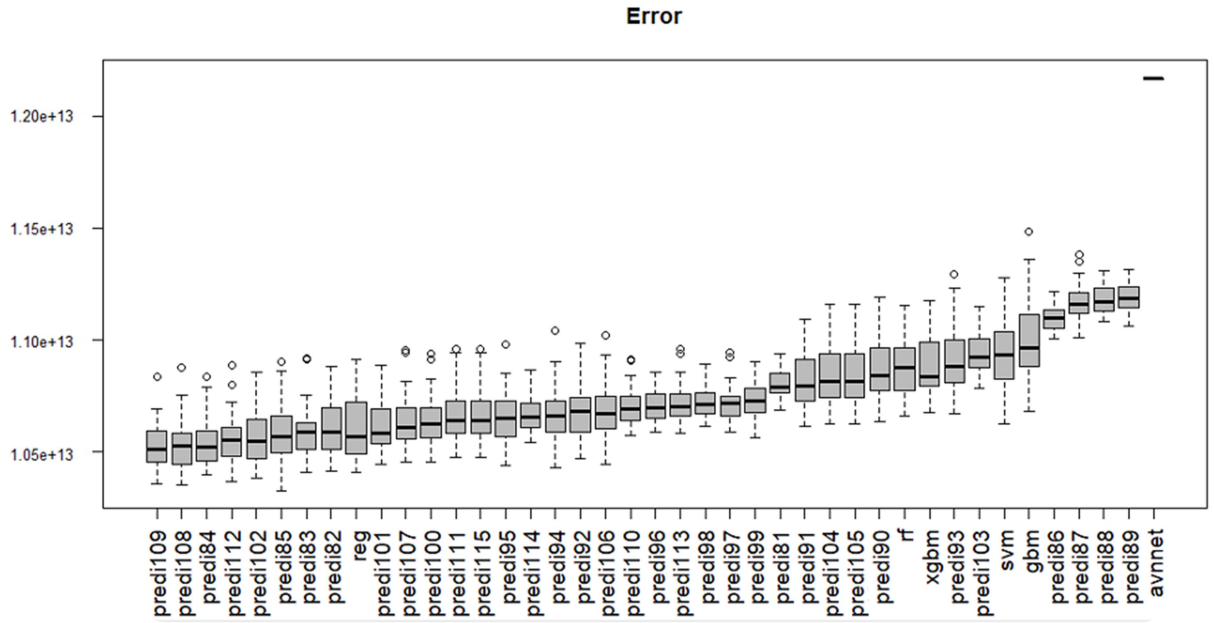


Figura 55. Resultados ensamblado con R parte 2.

4.7 Modelización variable objetivo continua coste total

Utilizando la totalidad de los datos y aplicando un modelo "clásico" con variable objetivo continua. Se predecirá el coste total por medio de los diferentes algoritmos. Para esto se realiza el mismo proceso de selección de variables con Enterprise Miner, Importancia de la variable y selección aleatoria, obteniendo los conjuntos de variables que se muestran en la Tabla 39.

Tabla 39. Conjuntos de variables seleccionadas.

Conjunto 1	Conjunto 2	Conjunto 3
Selección Miner	Importancia de la variable	Selección aleatoria
edad2	edad_M	edad
edad	edad	edad2
oncologia_adultos	edad2	dias_afiliacion
TI_enf_totales2	TI_G_enf_totales1	TI_G_enf_totales1
reumatologia_colageno	TI_enf_totales2	TI_enf_totales2
dialisis	VIH	genero_F
TI_G_enf_totales1	dialisis	TI_tipo2
	oncologia_adultos	VIH
		dialisis
		menos1
		oncologia_adultos
		reumatologia_colageno
		zona_2
		TI_OPT_edad4

4.7.1 Regresión

Se prueban cada uno de los conjuntos en el programa R, con validación cruzada repetida 200 veces, los resultados se muestran en la Tabla 40.

Tabla 40. Resultados Regresión Lineal.

Modelo	Conjunto de variables	RMSE	R cuadrado	#Variables
RL1	Selección miner	2021704	0.1952034	7
RL2	Importancia	2016950	0.1983241	8
RL3	Selección aleatoria	2021704	0.1952034	14

Se selecciona como mejor modelo de regresión el obtenido con el conjunto de variables "importancia", tiene el menor error cuadrático medio. En la Figura 56 se muestran los resultados gráficamente.

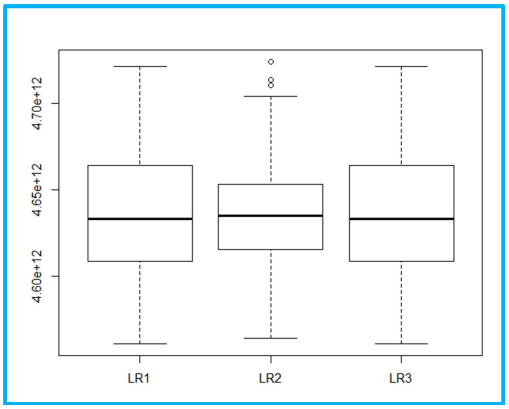


Figura 56. Resultados regresión.

4.7.2 Redes neuronales

Para definir el número de nodos a utilizar en la red, se realizan pruebas con validación cruzada repetida con diferentes valores iniciando en 3 como mínimo, incrementando de a dos hasta llegar al máximo que coincide con el número de nodos calculados para conjunto con la fórmula descrita en los apartados anteriores. Utilizando la función Caret, que permite probar diferentes valores para los nodos (size) y learning rate o decay de 0.01, 0.1 y 0.001. Se utiliza como máximo número de iteraciones de 200 y 10 repeticiones para cada combinación. Las configuraciones probadas y sus resultados se pueden ver en la Tabla 41.

Tabla 41. Configuraciones y resultados redes.

Modelo	Conjunto de variables	# Nodos	Tasa de aprendizaje (decay)	ERROR	RMSE	# Variables
red	Selección Miner	9	0,1	5,46277E+18	2205597	7
red2	Importancia	9	0,1	5,46277E+18	2205597	8
red3	Aleatoria	9	0,1	5,46277E+18	2205597	14

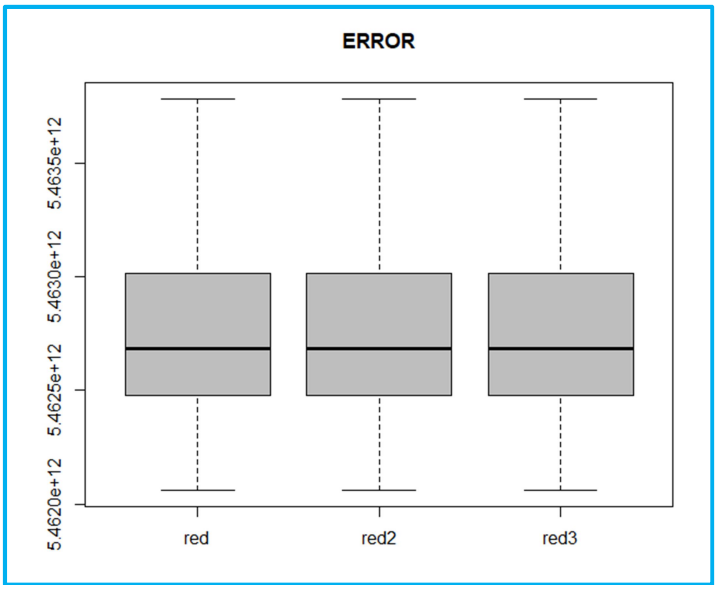


Figura 57. Resultados redes.

Debido a que los tres conjuntos probados arrojan los mismos resultados, se prefieren modelos más sencillos por lo cual se selecciona como modelo ganador el que menos variables utiliza, en este caso el conjunto selección Miner con siete variables.

4.7.3 Bagging y Random Forest

Inicialmente, se analiza y se define si se requiere parada anticipada (número de iteraciones máximas) para cada conjunto de variables (ver detalle en el Anexo XIII). Se prueban diferentes valores de tamaño de la muestra (sampsize): 1000, 2000 y 3000, con el modelo ganador se probará cambiando el número de observaciones mínimas por nodo, las configuraciones probadas se muestran en la Tabla 42.

Tabla 42. Configuraciones y resultados Bagging.

Modelo	Conjunto de variables	# Variables	# iteraciones	observaciones minimas por nodo	Tamaño de la muestra	RMSE	R cuadrado
Bag	Miner	7	1500	30	1000	2141474	0.1395399
Bag2	Importancia	8	1200	30	1000	2136861	0.1408355
Bag3	Aleatoria	14	1000	30	1000	2134821	0.1425156
Bag4	Miner	7	1500	30	2000	2134449	0.145315
Bag5	Importancia	8	1200	30	2000	2131232	0.1459228
Bag6	Aleatoria	14	1000	30	2000	2133654	.1449063
Bag7	Miner	7	1500	30	3000	2135555	0.1454396
Bag8	Importancia	8	1200	30	3000	2136185	0.1441584
Bag9	Aleatoria	14	1000	30	3000	2144523	0.1415518
Bag10	Importancia	8	1200	20	2000	2134956	0.1440427

Como se muestra en la Figura 58, el modelo Bag5 es el que presenta menor error se utilizará como mejor modelo.

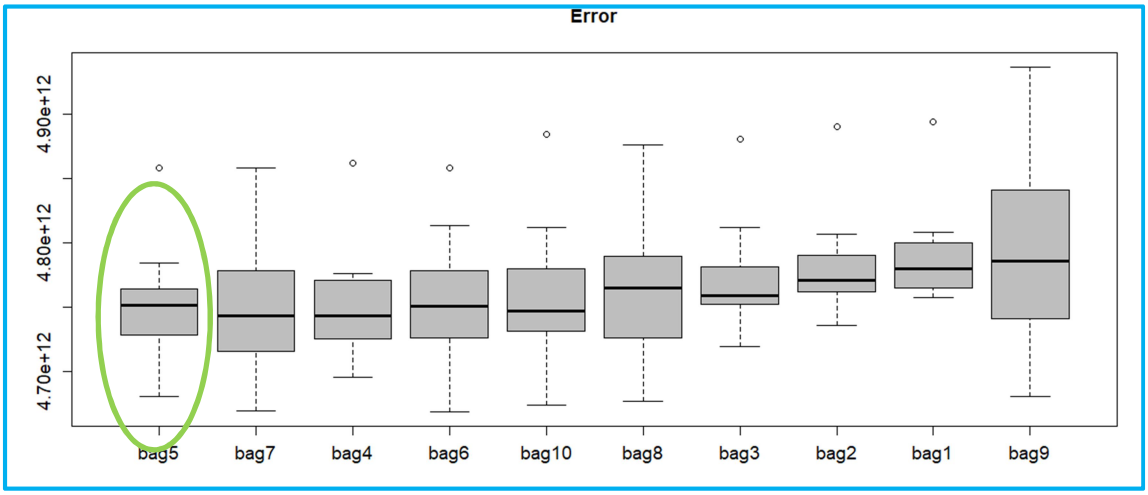


Figura 58. Resultados Bagging.

Dado que random forest es una generalización del bagging, se utilizarán los parámetros tuneados de observaciones mínimas por nodo, tamaño de la muestra, y número de iteraciones obtenidos por Caret y se modificará el número de variables a sortear que se utiliza en el modelo. En la Tabla 43, se relacionan las configuraciones probadas y resultados para random forest. En la Figura 59 se puede observar que el modelo RF10 presenta el menor error cuadrático medio.

Tabla 43. Configuraciones Random forest.

Modelo	Conjunto de variables	# Variables	# iteraciones	observ. Min. por nodo	Tamaño de la muestra	RMSE	R cuadrado
RF1	Miner	6	1500	30	2000	2134336	0.1451702
RF2	Miner	5	1500	30	2000	2134523	0.145026
RF3	Miner	4	1500	30	2000	2135108	0.1447818
RF4	Importancia	8	1200	30	2000	2131153	0.1457604
RF5	Importancia	6	1200	30	2000	2130654	0.1458348
RF6	Importancia	4	1200	30	2000	2131418	0.1453984
RF7	Aleatoria	12	1000	30	2000	2132794	0.1452277
RF8	Aleatoria	10	1000	30	2000	2132293	0.1453159
RF9	Aleatoria	8	1000	30	2000	2131136	0.1458105
RF10	Aleatoria	6	1000	30	2000	2130496	0.1459884
RF11	Aleatoria	4	1000	30	2000	2131465	0.1453235

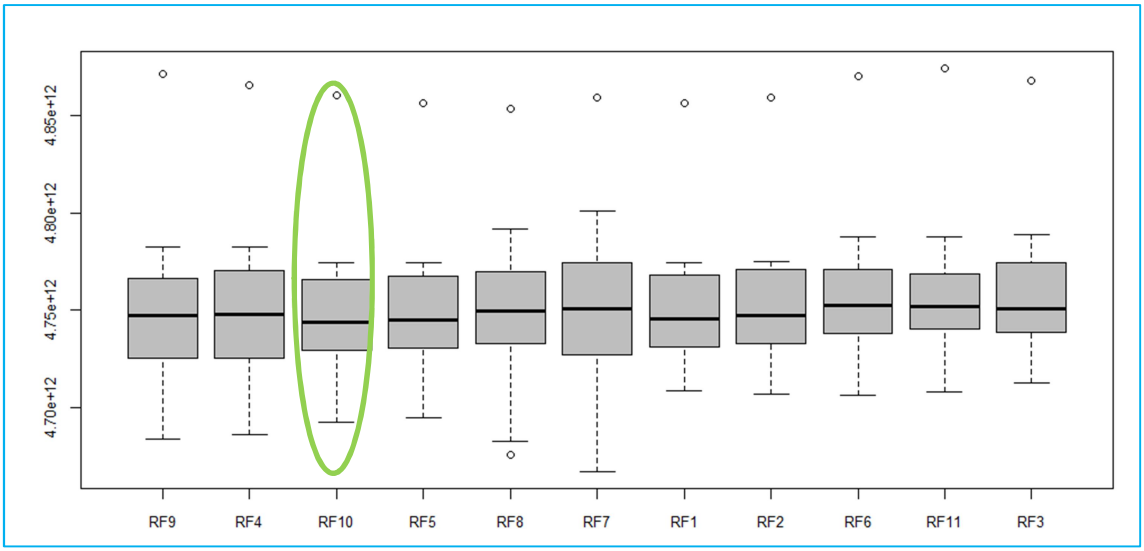


Figura 59. Resultados random forest

4.7.4 Incremento gradiente

Se utiliza Caret de R, con validación cruzada repetida, para probar los diferentes valores: shrinkage=0.001, 0.01, 0.05, 0.1, mínimo de observaciones por nodo 10, 20 y 30, número de iteraciones 1000, 2000 y 3000. Los valores óptimos de estos parámetros, serán aquellos valores que logren minimizar el error cuadrático medio. Los resultados de Caret se pueden ver en el Anexo XIV. En la Tabla 44 se relacionan las configuraciones probadas.

Tabla 44. Configuraciones Incremento gradiente.

Modelo	Conjunto de variables	Cte de regularización	# iteraciones	observaciones minimas por nodo	Semilla	RMSE	R cuadrado
GMB	Miner	0.01	1000	30	12345	2157695	0.1252085
GBM2	Importancia	0.001	1200	10	12345	2160587	0.1254387
GMB3	Aleatoria	0.01	1000	20	12345	2153106	0.1355125

En la Figura 60 se puede ver los resultados del incremento gradiente, se selecciona el modelo GBM3 como ganador, tiene el menor error cuadrático medio, aunque presenta mayor variabilidad que los demás modelos, esta variabilidad no es significativa.

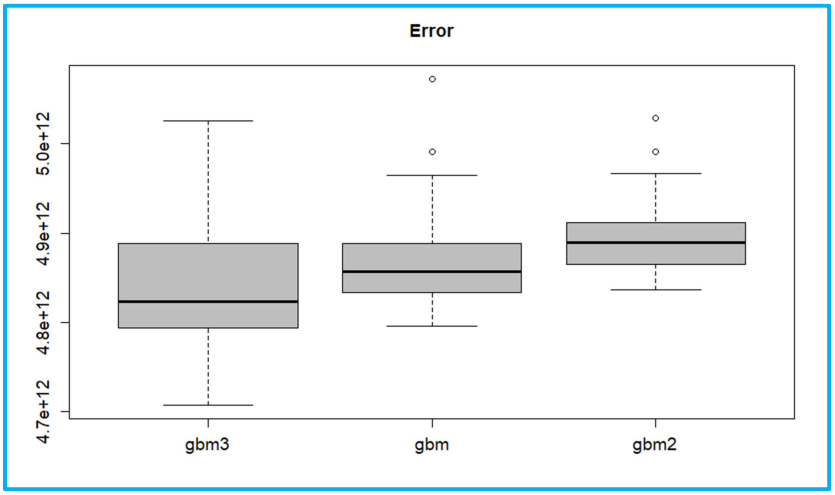


Figura 60. Resultado incremento gradiente.

4.7.5 XGBoost

Para determinar los valores de los parámetros se ejecuta Caret de R para cada conjunto de variables, probando las configuraciones sugeridas en (Brownlee, 2016) constante de regularización: 0.01, 0.015, 0.025, 0.05 y 0.1, número de observaciones mínimas por nodo: 5, 10, 15 , 20 y 30, número de iteraciones: 500, 1000, 2000 y 3000, máxima profundidad de los árboles: 5, 7, 9 y 12, penalización gamma 0, 0.3, 0.5, 0.7 y 1, con sorteo de variables y observaciones de 0.8 y sin sorteo de ambas. También se realizan otras validaciones como por ejemplo, si se requiere parada anticipada (early stopping) y análisis de valor óptimo para el parámetro lambda. Ver los resultados detallados en Anexo XV. Se prueba modificando la semilla de la validación cruzada, el parámetro gamma, sorteo de variables y sorteo de observaciones.

En la Tabla 45, se relacionan las configuraciones probadas y resultados para xgboost, el mejor modelo se obtiene con el conjunto de variables aleatoria 2, modelo XGBM8. En la Figura 44 se puede observar que es el modelo que menor error presenta.

Tabla 45.Configuraciones y resultados xgboost.

Modelo	Conjunto de variables	Cte de regularización	# iteraciones	observaciones minimas por nodo	Semilla	gamma	sorteo de variables	lambda	sorteo de observaciones	prof. máxima	RMSE	R cuadrado
XGBM1	Miner	0.05	300	30	1234	0	no		no	5	2187629	0.1192303
XGBM2	Importancia	0.01	300	30	12345	0	no		no	5	2143963	0.1371028
XGBM3	Aleatoria	0.01	300	30	12345	0	no		no	5	2143963	0.1371028
XGBM4	Importancia	0.01	300	20	12345	0	no		no	5	2172627	0.1189787
XGBM5	Importancia	0.01	300	30	12345	0	0.8		no	5	2143531	0.137332
XGBM6	Importancia	0.01	300	30	12345	0	0.8	10	no	5	2141245	0.1372249
XGBM7	Importancia	0.01	300	30	12345	0	0.8		0.8	5	2138490	0.140265
XGBM8	Importancia	0.01	300	30	12345	0	0.8		0.8	10	2138664	0.140211

En la Figura 61 se puede observar que el modelo XGBM7 presenta los mejores resultados en cuanto al error cuadrático medio.

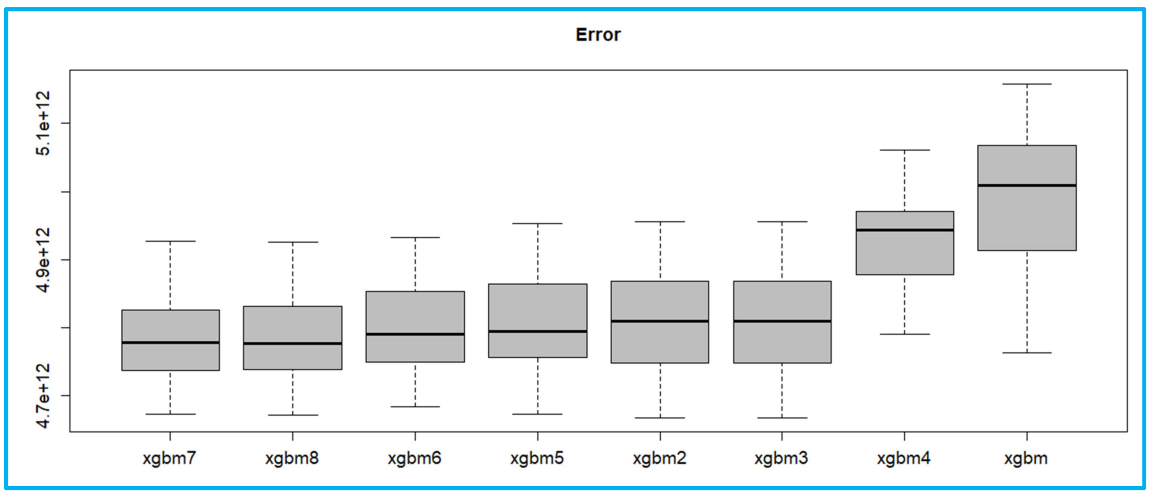


Figura 61. Resultados xgboost.

4.7.6 Máquinas de soporte vectorial (SVM)

4.7.6.1 SVM Lineal

Para las máquinas de soporte vectorial lineales, se requiere definir la constante de regularización C, se utiliza de nuevo Caret de R para probar diferentes valores de

esta constante, como son: 0.001, 0.01, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 7, 10, 12, 20, 40, 60, 80 y 100 para cada conjunto de variables. Se prueban por validación cruzada repetida todas las configuraciones obtenidas con Caret. Las configuraciones y resultados se muestran en la Tabla 46.

Tabla 46. Configuraciones y resultados SVM lineal.

Modelo	Conjunto de variables	Cte de regularización	Semilla	RMSE	R cuadrado
SVML	Miner	10	1234	2146149	0.1436253
SVML2	Importancia	0.01	1234	2145853	0.1440073
SVML3	Aleatoria1	5	12345	2145232	0.1444387

El modelo SVML3 presenta el menor error cuadrático medio de los tres conjuntos probados, aunque tiene mayor variabilidad que el modelo SVML2 la diferencia no es significativa.

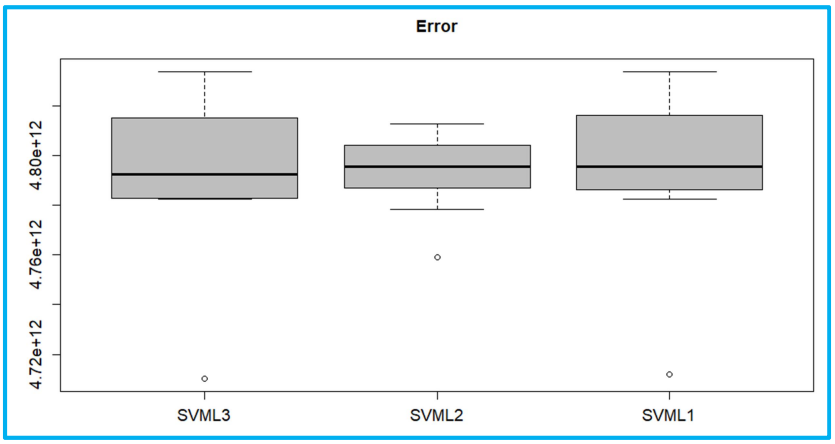


Figura 62. Resultados SVM lineal.

4.7.6.2 SVM Radial

El RBF requiere que se defina el valor de la constante de regularización y sigma, de nuevo se utiliza Caret para probar: C=0.01, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, y 10 y sigma= 0.01, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10 y 30. Se prueban por validación cruzada repetida 200 veces las configuraciones obtenidas con Caret que se relacionan en la Tabla 47.

Tabla 47. Configuraciones y resultados SVM Radial.

Modelo	Conjunto de variables	Cte de regularización	Sigma	Semilla	RMSE	R cuadrado
SVMRBF1	Miner	10	0.2	1234	2099778	0.1840146
SVMRBF2	Importancia	10	0.1	1234	2121332	0.1626621
SVMRBF3	Aleatoria1	10	0.01	1234	2121376	0.1594006

Como se puede observar en la Figura 63, el modelo SVMRBF1 es el que tiene el mejor resultado.

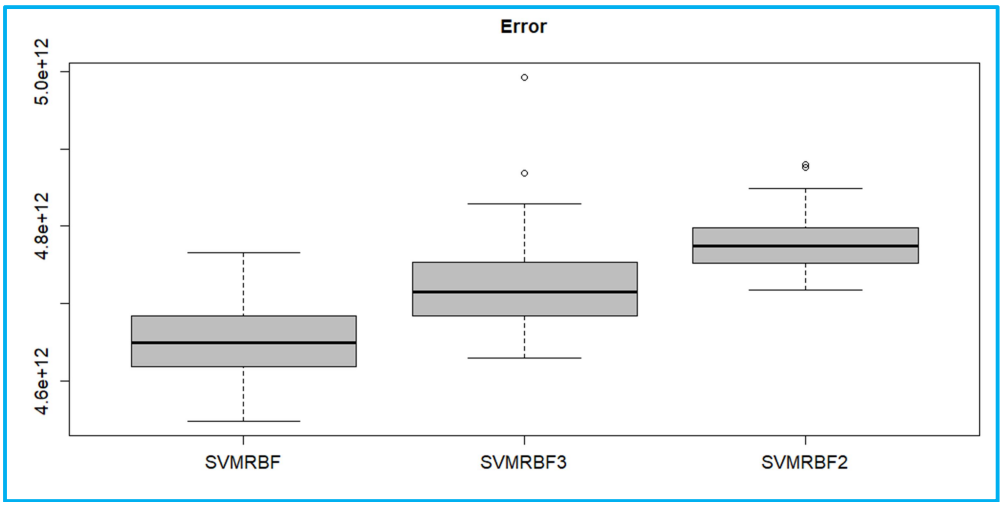


Figura 63. Resultados SVM Radial.

4.7.7 Evaluación de los modelos

Una vez se tienen las configuraciones óptimas de cada algoritmo, se corren de nuevo con validación cruzada repetida, variando la semilla 100 veces y se comparan los resultados. Como se puede ver en la Figura 64, la regresión es la que tiene menor error cuadrático medio, la diferencia con los algoritmos basados en arboles no es muy significativa. De nuevo la red no logra competir con los demás algoritmos. Se realizarán pruebas de ensamblado con estos modelos ganadores.

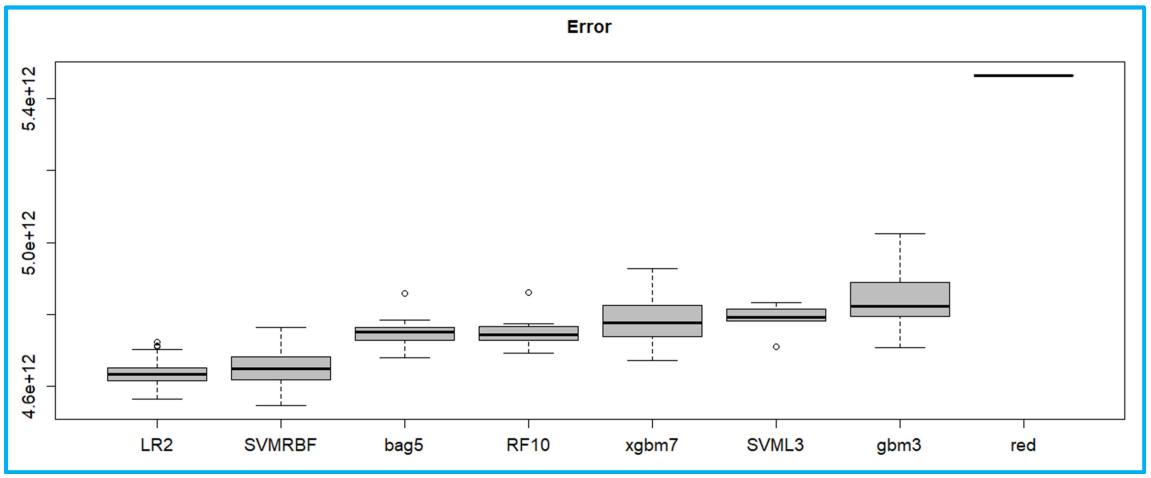


Figura 64. Comparación de los mejores modelos.

4.7.8 Ensamblado

En la Tabla 48 se listan las combinaciones de modelos utilizadas para ensamble.

Tabla 48. Configuraciones ensamblado.

Modelo	Ensamble	Modelo	Ensamble
predi81	reg+avnnet	predi99	reg+avnnet+svm
predi82	reg+rf	predi100	reg+rf+gbm
predi83	reg+gbm	predi101	reg+rf+xgbm
predi84	reg+xgbm	predi102	reg+rf+svm
predi85	reg+svm	predi103	rf+avnnet+gbm

predi86	avnnet+rf	predi104	rf+gbm+xgbm
predi87	avnnet+gbm	predi105	rf+gbm+xgbm
predi88	avnnet+xgbm	predi106	svm+gbm+xgbm
predi89	avnnet+svm	predi107	reg+gbm+xgbm
predi90	rf+gbm	predi108	reg+gbm+svm
predi91	rf+xgbm	predi109	reg+xgbm+svm
predi92	rf+svm	predi110	reg+rf+gbm+avnnet
predi93	gbm+xgbm	predi111	reg+rf+gbm+xgbm
predi94	gbm+svm	predi112	reg+svm+gbm+xgbm
predi95	xgbm+svm	predi113	reg+avnnet+gbm+xgbm
predi96	reg+avnnet+rf	predi114	reg+avnnet+svm+xgbm
predi97	reg+avnnet+gbm	predi115	reg+rf+gbm+xgbm
predi98	reg+avnnet+xgbm		

En la Figura 65 se muestran los resultados del ensamblado, el modelo predi84 de ensamble que incluye regresión, xgboost, es el que logra el mejor resultado. Cabe destacar que de nuevo la diferencia con respecto a la regresión es mínima y dado que se sacrifica interpretabilidad del modelo se selecciona la regresión como modelo ganador.

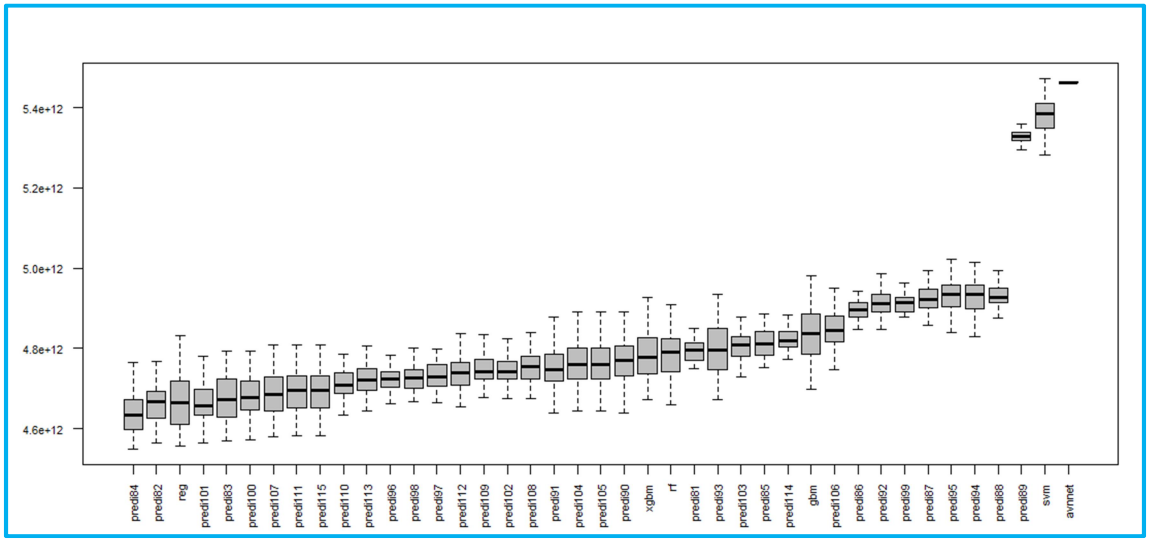


Figura 65. Resultados ensamblado.

4.8 Evaluación final de los modelos

Se probaron 357 modelos de predicción resumidos en la Tabla 49. Se abordó el problema aplicando dos estrategias de predicción, la primera que consistió en dividir en dos partes la modelación y para la segunda se aplicaron algoritmos de predicción "clásicos" al conjunto de datos completo.

Tabla 49. Resumen modelos probados.

Técnica	Primera parte	Segunda parte	Modelo completo
Regresión	9	8	3
Redes neuronales	29	8	3
Bagging	20	12	10
Random forest	22	12	11
incremento gradiente	24	5	3
Xgboost	16	12	8
Máquinas de soporte vectorial	10	8	6
Ensamblado	48	35	35
Total	**178**	**100**	**79**

Modelo de dos partes: Para el cálculo del modelo final, una vez se define qué modelos se van a utilizar en cada parte, se usa la fórmula:

$$E(Y|X) = P\,(Y > 0|X) * E(Y|X, Y > 0)$$

Donde la expresión $P\,(Y > 0|X)$, corresponde a la probabilidad de ocurrencia del evento y $E(Y|X, Y > 0)$ es el valor esperado del coste dada la probabilidad de que este sea mayor que cero.

Calculo medidas de adecuación

Se realiza la predicción para cada observación, y se calculan las medidas de evaluación, se utilizará el error cuadrático medio:

$$MSEp_j = \frac{\sum_{i=1}^{Nobs}\left(y_i - \hat{y}_{ij}\right)^2}{Nobs}$$

Y el indicador de ajuste financiero global, definido como la diferencia entre el coste total observado para la población y el coste total predicho por el modelo en términos relativos:

$$IF_p = \frac{\sum_{i=0}^{Nobs}\left(\hat{y}_{ij} - y_i\right)}{\sum_{i=1}^{Nobs} y_i}$$

En la Tabla 50 se muestran los resultados de las medidas de evaluación del modelo de dos partes y el modelo completo. Ambos modelos presentan un R cuadrado inferior al 0.2. Este valor indica que el modelo no logra explicar de forma individual la variación del coste del servicio de salud, este resultado se esperaba dado que predecir individualmente el coste incluso en el caso de que dos personas presenten exactamente las mismas características, alguno puede presentar un siniestro y generar costes altos mientras que el otro puede que no genere ningún coste. Desde un punto de vista financiero y de planeación la entidad promotora de salud no le

interesa conocer el coste que genera cada persona sino el total, ya que la gestión financiera se realiza a nivel global y no individual. Teniendo en cuenta esto la medida de evaluación más apropiada es el indicador de ajuste financiero global, que permite evaluar que tanto se ajusta el modelo al coste real.

En cuanto a este indicador financiero el modelo de dos partes está subestimando el costo en un 1%, mientras que el modelo completo logra acertar en la predicción del costo con una sobre valoración inferior al 2%. Se seleccionará el modelo completo para la predicción del coste total ya que requiere menor esfuerzo en su elaboración y logra tener un resultado que le permite a la entidad no presentar déficit financiero.

Tabla 50. Medidas de evaluación de los modelos.

Medida	Modelo dos partes	Modelo completo
MSE	4,4409E+12	4,5514E+12
RSQUARE	0.186	0.1758
IF	-1%	1,9%

El modelo ganador utiliza el conjunto de variables "importancia" que cuenta con ocho variables de entrada: tres continuas y cinco categóricas. Con respecto a las variables utilizadas y los coeficientes del modelo de la Tabla 51, se puede concluir que, en términos generales, tanto para hombres como mujeres el coste incrementa con la relación cuadrática de la edad. No obstante, debido al coeficiente negativo de la interacción entre el sexo y la edad, se puede concluir que las mujeres presentarán mayor coste que los hombres a medida que la edad incrementa.

La variable TI_G_enf_totales1, que toma el valor 1 si la persona presenta alguna de las enfermedades de alto costo consideradas (sin importar cuál específicamente), muestra que se incrementa el coste en 1.6 millones de pesos, cuando una persona presenta una o más enfermedades y se incrementa en 2.7 millones de pesos cuando presenta dos o más (TI_enf_totales2). Así mismo, si el afiliado tiene VIH, su coste se incrementará en 11 millones de pesos; diálisis representa un incremento de casi 17 millones de pesos y, por último, un afiliado con cáncer incrementa el coste en 235 mil pesos.

Tabla 51. Coeficientes modelo de regresión.

Variable	Coeficientes
(Intercept)	424.399
edad_M	-2.186
edad	-16.816
edad2	337
TI_G_enf_totales1 (1)	1.620.041
TI_enf_totales2 (1)	2.738.731
VIH (1)	11.115.028
dialisis (1)	16.816.212
oncologia_adultos (1)	235.125

En la Figura 66, se puede observar los resultados de la regresión lineal. Las variables que tienen más impacto en el coste total en orden son: diálisis y VIH.

```
Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)         4.244e+05  7.311e+04   5.805 6.69e-09 ***
edad_M             -2.186e+03  1.341e+03  -1.630 0.103091
edad               -1.682e+04  4.462e+03  -3.769 0.000165 ***
edad2               3.375e+02  5.805e+01   5.814 6.33e-09 ***
TI_G_enf_totales11  1.620e+06  9.078e+05   1.785 0.074376 .
TI_enf_totales21    2.739e+06  8.263e+05   3.315 0.000922 ***
VIH1                1.112e+07  6.257e+05  17.764  < 2e-16 ***
dialisis1           1.682e+07  1.005e+06  16.726  < 2e-16 ***
oncologia_adultos1  2.351e+05  4.230e+05   0.556 0.578352
```

Figura 66. Resultado regresión lineal.

5. Conclusiones y recomendaciones

El objetivo del presente trabajo es predecir el coste total anual del servicio de salud en una entidad promotora de salud, incluyendo las posibles variables que afectaban la predicción. Si bien el R cuadrado es bajo, esto confirma que el coste médico no es fácil de predecir por persona dadas las características particulares del fenómeno que presenta alta variabilidad entre individuos con las mismas características.

No se considera de utilidad para la entidad el conocimiento del coste generado por cada persona afiliada dado que la planeación financiera se realiza a nivel global. Los resultados obtenidos con el modelo completo logran predecir este coste total con un error del 2%.

Se concluye que, para este caso, el modelo de dos partes logra mejorar ligeramente los resultados pero requiere un esfuerzo adicional comparado con el modelo completo.

Todas las técnicas de predicción empleadas generaron resultados similares, esto significa que se puede usar cualquiera de ellas con el objetivo de predecir el coste total. La selección de la técnica a utilizar dependerá del conocimiento estadístico y del fenómeno que tenga el analista que aplique las técnicas y de las necesidades de interpretación de las variables que afectan el coste.

En cuanto a las variables que más influyen en el coste total que hacen referencia a enfermedades pre-existentes se concluye que los afiliados que requieren diálisis son los que más coste generan para la entidad, así como aquellos pacientes que han sido diagnosticados con VIH. En cuanto a las variables que tienen que ver con las características de la población, la edad al cuadrado es la variable más importante a la hora de predecir.

Por último, en el caso de no requerir interpretación de las variables que afectan el coste se recomienda utilizar los modelos de ensamblado que logran mejorar la predicción.

Como trabajo futuro sería interesante utilizar el indicador de ajuste financiero global en la selección de los mejores modelos con cada técnica, ya que esta medida de evaluación es más apropiada para el problema planteado en este trabajo.

6. Bibliografía

Adams, D.R., Harrell, F.E., Richard Smith, L., Mark, D.B., Califf, R.M., Pryor, D.B., Glower, D., Lipscomb, J., Hlatky, M., 1993. Comparison of analytic models for estimating the effect of clinical factors on the cost of coronary artery bypass graft surgery. J. Clin. Epidemiol. 46, 261–271. https://doi.org/10.1016/0895-4356(93)90074-B

Brownlee, J., 2016. How to Configure the Gradient Boosting Algorithm. Mach. Learn. Mastery. URL https://machinelearningmastery.com/configure-gradient-boosting-algorithm/ (accessed 9.8.19).

Cragg, J.G., 1971. Some Statistical Models for Limited Dependent Variables with Application to the Demand for Durable Goods. Econometrica 39, 829–844. https://doi.org/10.2307/1909582

Diehr, P., Yanez, D., Ash, A., Hornbrook, M., Lin, D.Y., 1999. Methods for Analyzing Health Care Utilization and Costs. Annu. Rev. Public Health 20, 125–144. https://doi.org/10.1146/annurev.publhealth.20.1.125

Duan, N., Manning, W.G., Newhouse, J.P., Morris, C.N., 1982. A Comparison of Alternative Models for the Demand for Medical Care. RAND Corporation, Santa Monica, CA.

Friedman, J.H., Hastie, T., Tibshirani, R., 2009. The elements of statistical learning: data mining, inference, and prediction, 2nd ed.

Joyanes-Aguilar, L., Castaño, N.J., Osorio, J.H., 2015. [Simulation and data mining model for identifying and prediction budget changes in the care of patients with hypertension]. Rev. Salud Publica 17, 789–800. https://doi.org/10.15446/rsap.v17n5.39610

Lipscomb, J., Matchar, G.S.D.B., Hasselblad, P.V., Marek Ancukiewicz, 1998. Predicting the Cost of Illness: A Comparison of Alternative Models Applied to Stroke. Med Decis Mak. 1 18 2_suppl, S39–S56.

Liu, L., Strawderman, R.L., Cowen, M.E., Shih, Y.-C.T., 2010. A flexible two-part random effects model for correlated medical costs. J. Health Econ. 29, 110–123. https://doi.org/10.1016/j.jhealeco.2009.11.010

Manning, W.G., Mullahy, J., 2001. Estimating log models: to transform or not to transform? J. Health Econ. 20, 461–494. https://doi.org/10.1016/S0167-6296(01)00086-8

Mosquera, J., 2016. Estimación del Costo Medio Anual del Servicio de Salud para la población afiliada a una EPS en Colombia 2013. 14.

Mullahy, J., 1998. Much ado about two: reconsidering retransformation and the two-part model in health econometrics. J. Health Econ. 17, 247–281. https://doi.org/10.1016/S0167-6296(98)00030-7

Portela, J., 2019. Notas de clase Machine Learning.

Vargas, J.M., Giraldo, J.A., 2014. Modelo de Predicción de Costos en Servicios de Salud Soportado en Simulación Discreta. Inf. Tecnológica 25, 175–184. https://doi.org/10.4067/S0718-07642014000400019

7. Anexos

ANEXOS PARTE I VARIABLE OBJETIVO BINARIA

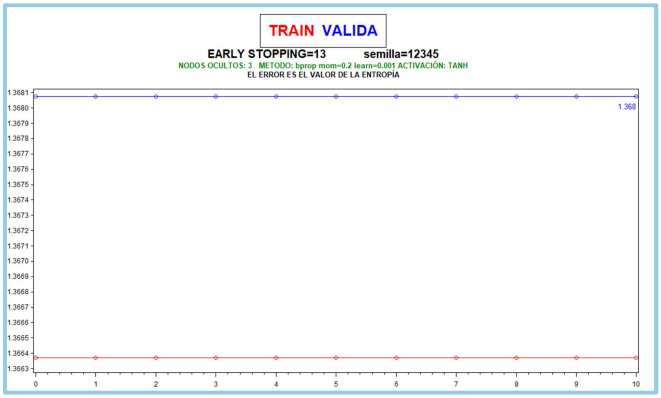
Anexo I. Pruebas regresión logística

Configuración mejores conjuntos, con n variables desde 6 hasta 10:

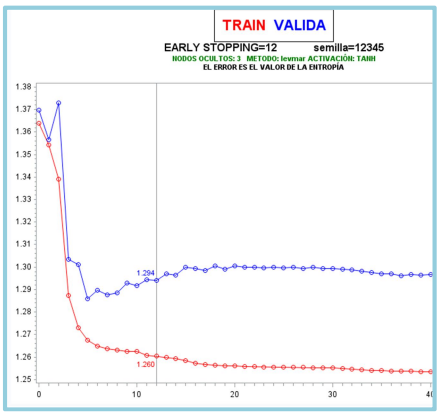
```
6     682.5275  edad_M TI_edad21 TI_G_enf_totales1 TI_dias_afil1 TI_tipo2 zona_9
7     714.4172  TI_G_enf_totales1 TI_dias_afil1 genero_F TI_tipo2 zona_9 TI_OPT_edad2 TI_OPT_edad3
8     739.7103  edad_F edad_M edad2 TI_G_enf_totales1 TI_dias_afil1 TI_tipo2 zona_9 TI_OPT_edad2
9     756.7567  edad_F edad_M edad2 TI_G_enf_totales1 TI_dias_afil1 TI_tipo2 zona_5 zona_9 TI_OPT_edad2
0     771.3328  edad_F edad_M edad2 TI_G_enf_totales1 TI_dias_afil1 TI_tipo2 zona_1 zona_5 zona_9 TI_OPT_edad2
```

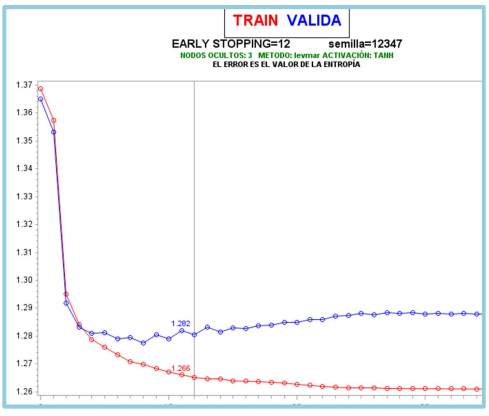
Anexo II. Análisis de parada anticipada para redes clasificación

Conjunto selección miner: Algoritmo bprop función de activación tanh con 3 nodos ocultos, momentum 0.2 y tasa de aprendizaje 0.001. No requiere parada anticipada.

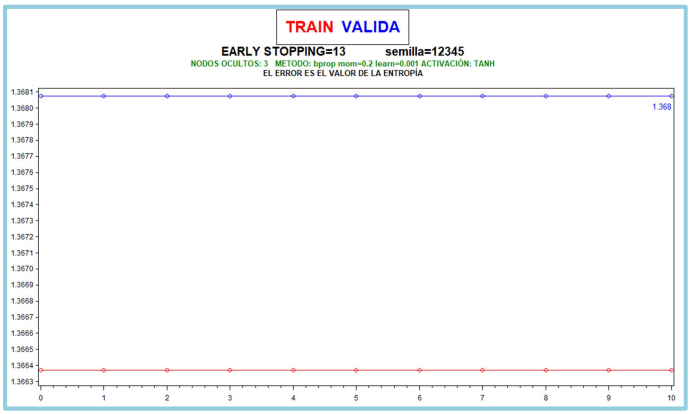


Algoritmo levmar función de activación tanh con 3 nodos ocultos, necesita parada anticipada en la 8 iteración, Se confirma la estabilidad del resultado cambiando la semilla aleatoria.

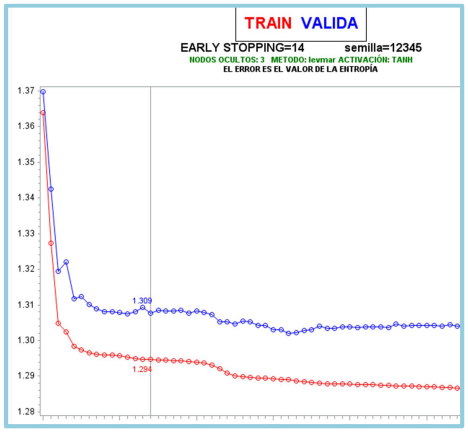


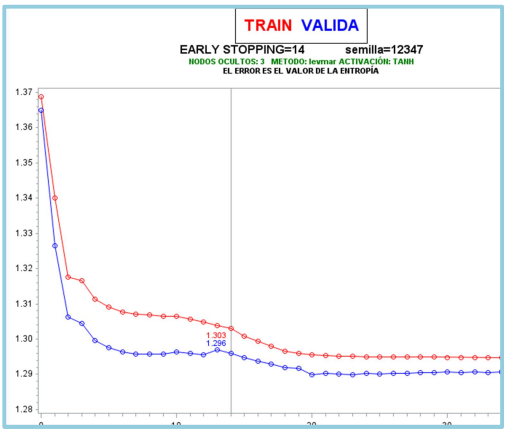


Conjunto importancia de la variable: Algoritmo bprop, función de activación tanh y 3 nodos momentum 0.2 y tasa de aprendizaje 0.001, no requiere parada anticipada el error es constante.

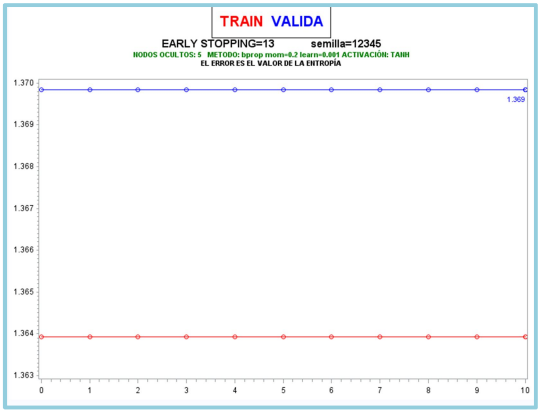


Algoritmo levmar función de activación tanh y 3 nodos, no se necesita parada anticipada, el error en validación disminuye con el incremento de las iteraciones. Se confirma la estabilidad del resultado cambiando la semilla aleatoria.



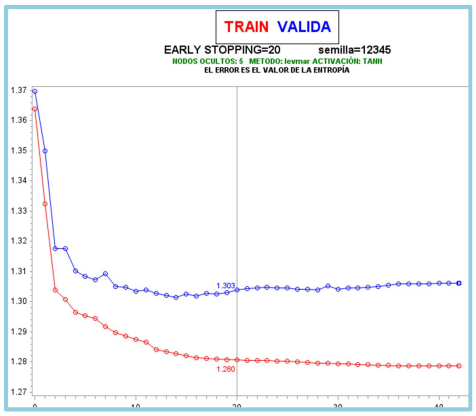


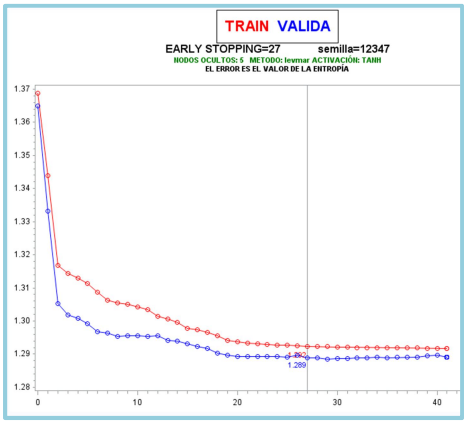
Importancia de la variable con 5 nodos: Con algoritmo Bprop, momentum 0.2, tasa de aprendizaje 0.001 función de activación Tanh.



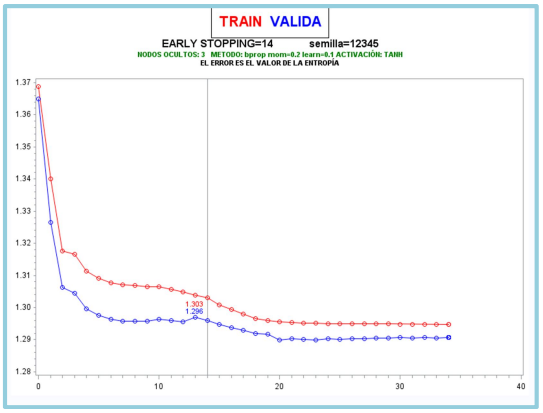
Con algoritmo Levmar, función de activación tanh y 5 nodos. Requiere parada anticipada en la iteración 20 el error incrementa un poco a partir de este punto

aunque no es muy significativo. Se confirma la estabilidad del resultado cambiando la semilla aleatoria. Se probará con y sin parada anticipada.

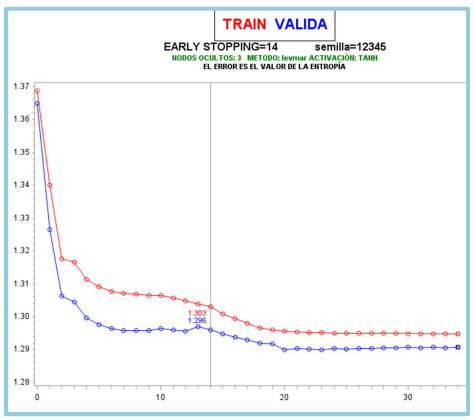


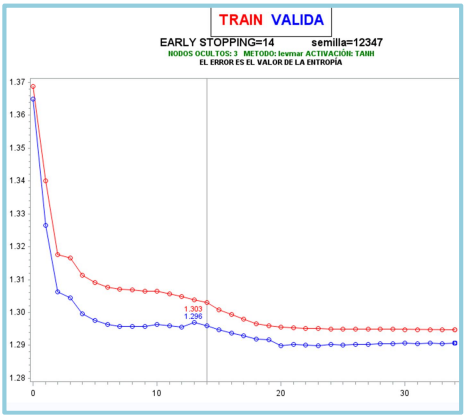


Conjunto Aleatoria 1: Algoritmo bprop, función de activación tanh, 3 nodos ocultos momentum 0.2 y tasa de aprendizaje 0.1, no requiere parada anticipada el error disminuye con el incremento de las iteraciones.



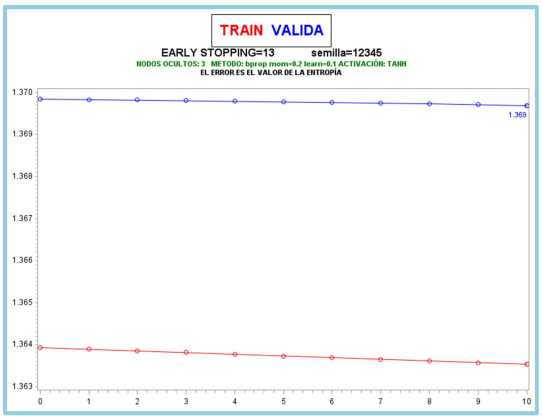
Con Algoritmo Levmar, función de activación tanh, 3 nodos ocultos. No requiere parada anticipada, no se incrementa el error con el incremento del número de iteraciones. Se confirma la estabilidad del resultado cambiando la semilla aleatoria.



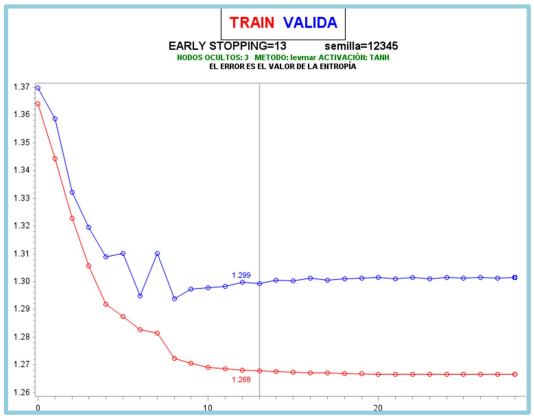


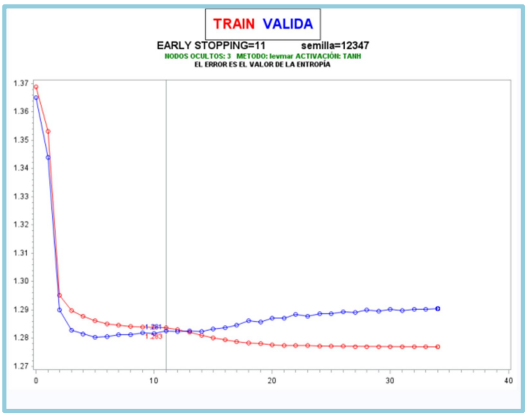
Conjunto Aleatoria 2 con 3 nodos: Algoritmo bprop, función de activación tanh, 3 nodos ocultos momentum 0.2 y tasa de aprendizaje 0.1, no requiere parada

anticipada el error disminuye con el incremento de las iteraciones aunque no significativamente.

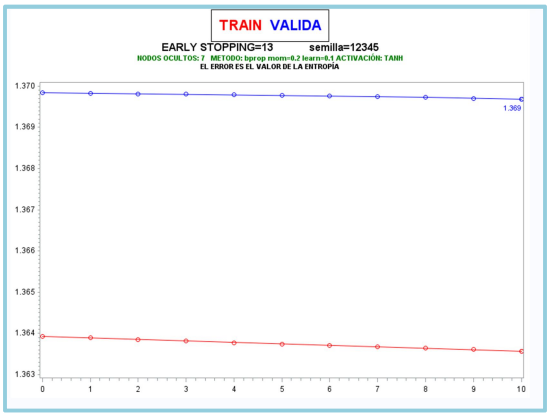


Con Algoritmo Levmar, función de activación tanh, 3 nodos ocultos. Requiere parada anticipada en la iteración 11. Se confirma la estabilidad del resultado cambiando la semilla aleatoria.

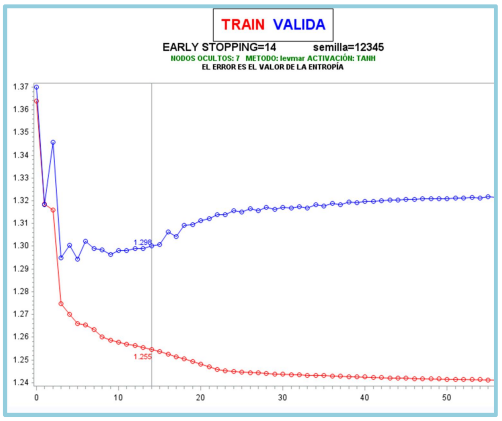


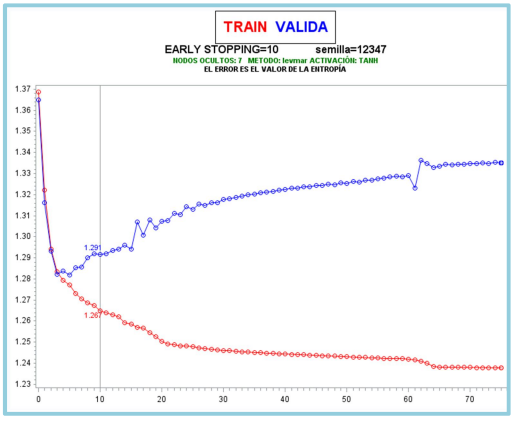


Aleatoria 2 con 7 nodos: Algoritmo bprop, función de activación tanh, 7 nodos ocultos momentum 0.2 y tasa de aprendizaje 0.1, no requiere parada anticipada el error disminuye con el incremento de las iteraciones aunque no significativamente.

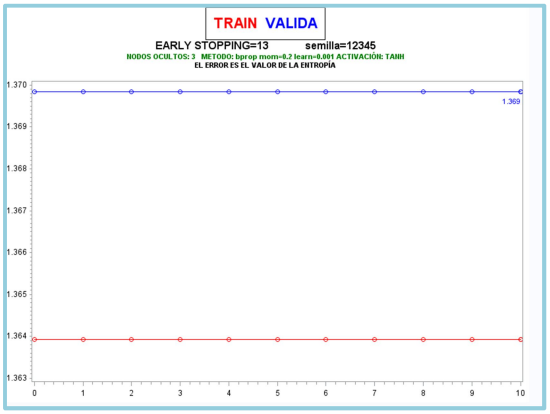


Con Algoritmo Levmar, función de activación tanh, 7 nodos ocultos. Requiere parada anticipada en la iteración 10. Se confirma la estabilidad del resultado cambiando la semilla aleatoria.

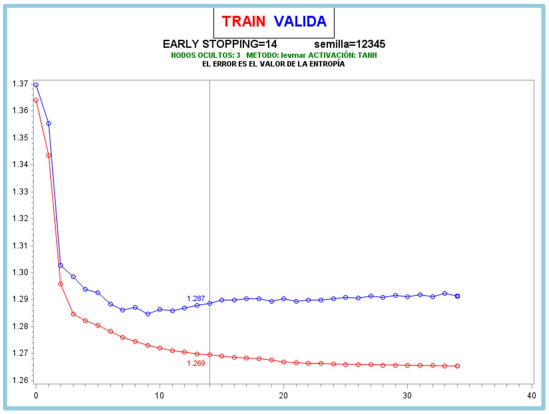


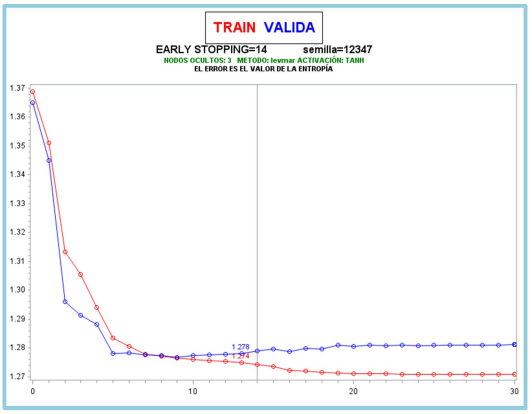


Conjunto mejor con 10: Algoritmo bprop, función de activación tanh, 3 nodos ocultos momentum 0.2 y tasa de aprendizaje 0.001, no requiere parada anticipada el error disminuye con el incremento de las iteraciones aunque no significativamente.



Con Algoritmo Levmar, función de activación tanh, 3 nodos ocultos. Requiere parada anticipada en la iteración 8. Se confirma la estabilidad del resultado cambiando la semilla aleatoria.

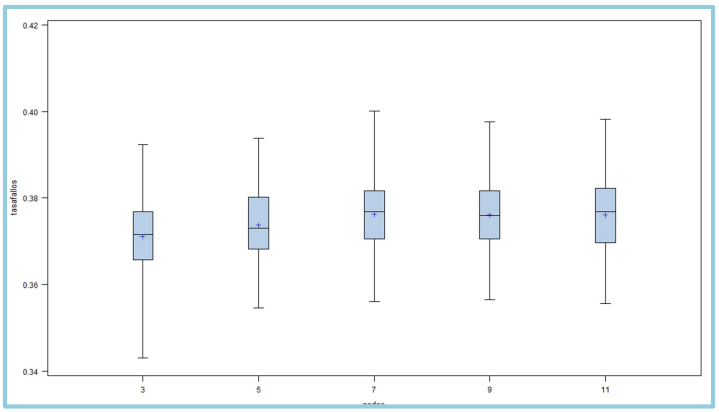




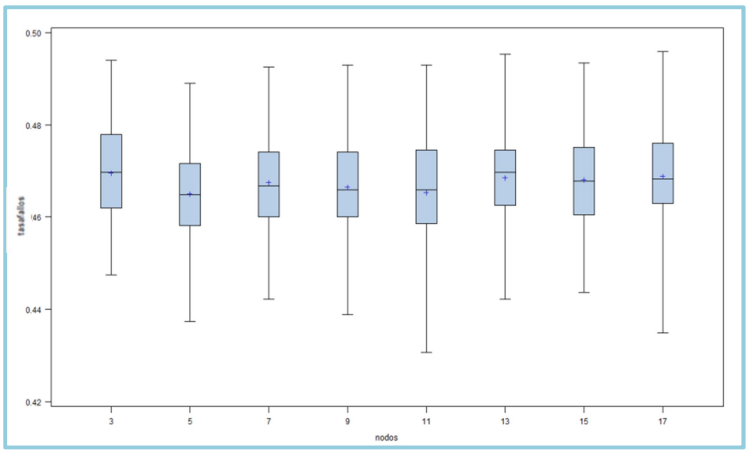
Anexo III. Definición número de nodos para redes clasificación con SAS Base

Se selecciona el número de nodos con el que se obtenga la menor tasa de fallos.

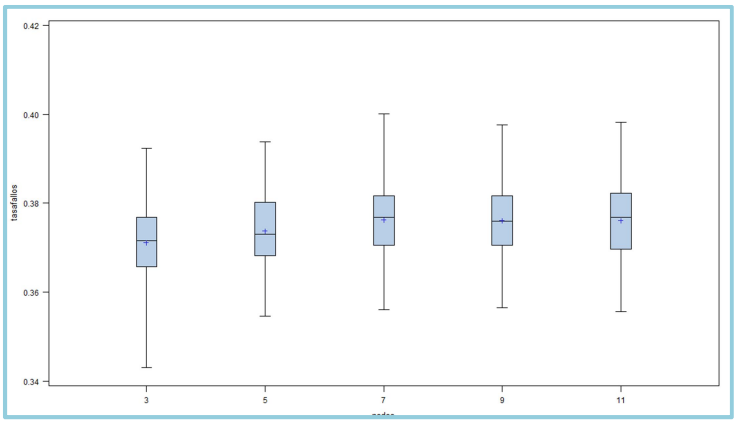
Conjunto Selección Miner: La menor tasa de fallos se consigue utilizando 3 nodos.



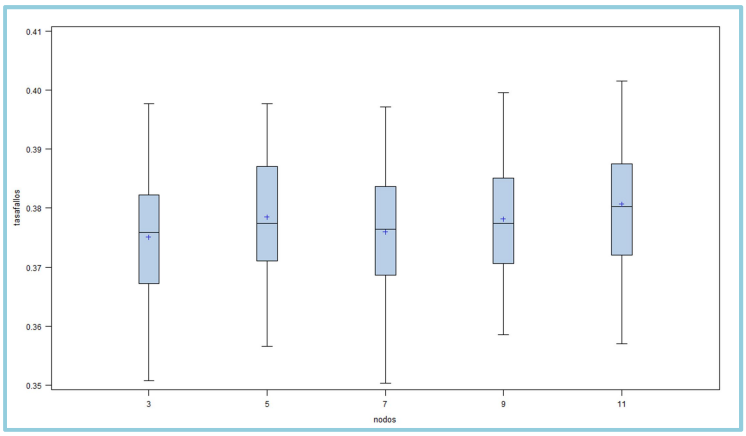
Conjunto importancia de la variable: La menor tasa de fallos se consigue con 5 nodos, dado que el resultado es diferente al obtenido en R, se probará con los dos valores. 3 nodos y 5 nodos.



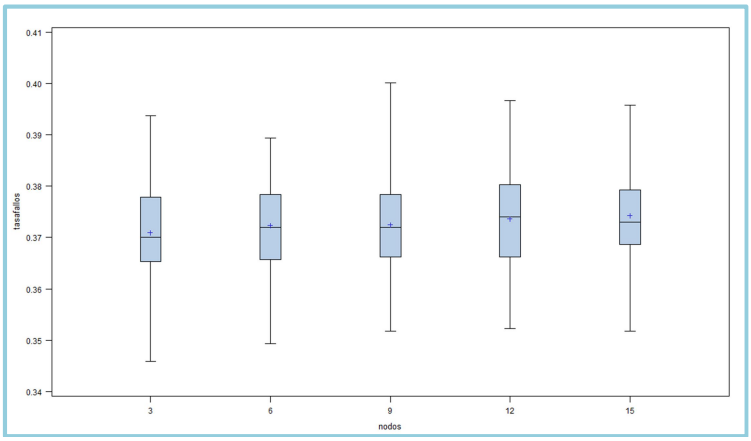
Conjunto Aleatoria 1: La menor tasa de fallos se consigue con 3 nodos.



Conjunto Aleatoria 2: La menor tasa de fallos se consigue con 3 nodos, sin embargo al utilizar 7 nodos el resultado es similar, se realizarán pruebas con los dos valores.

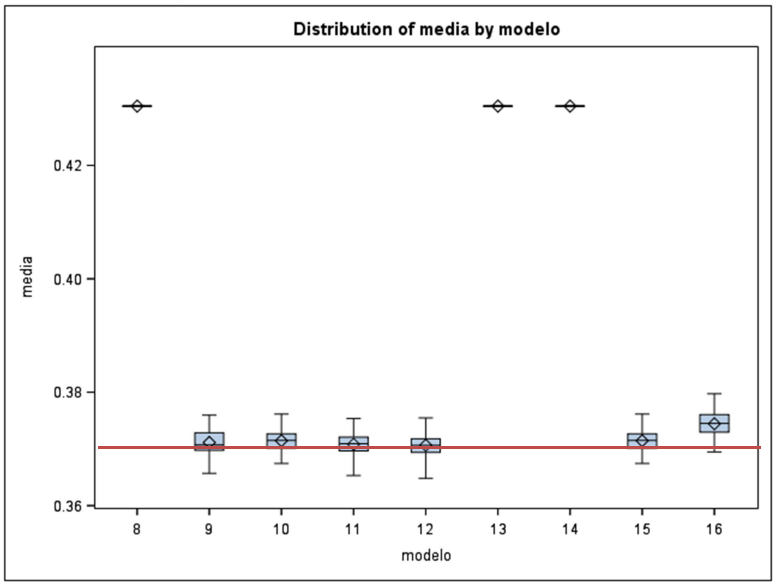


Conjunto Mejor con 10 variables: La menor tasa de fallos se consigue con 3 nodos.

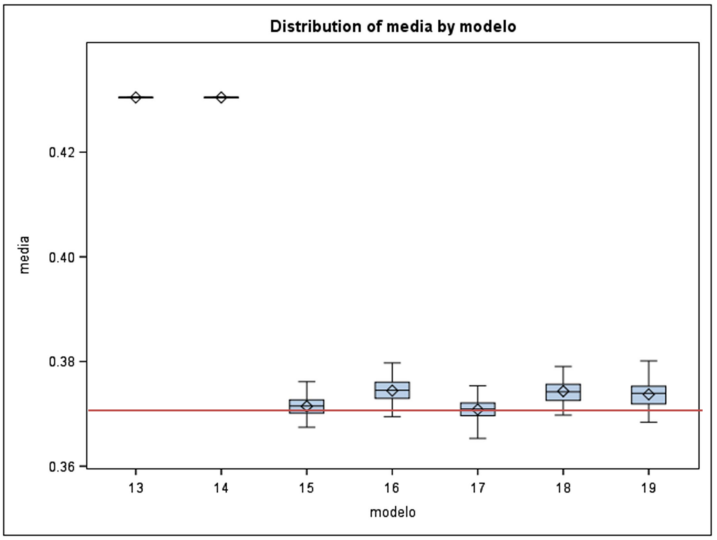


Anexo IV. Resultados redes clasificación SAS

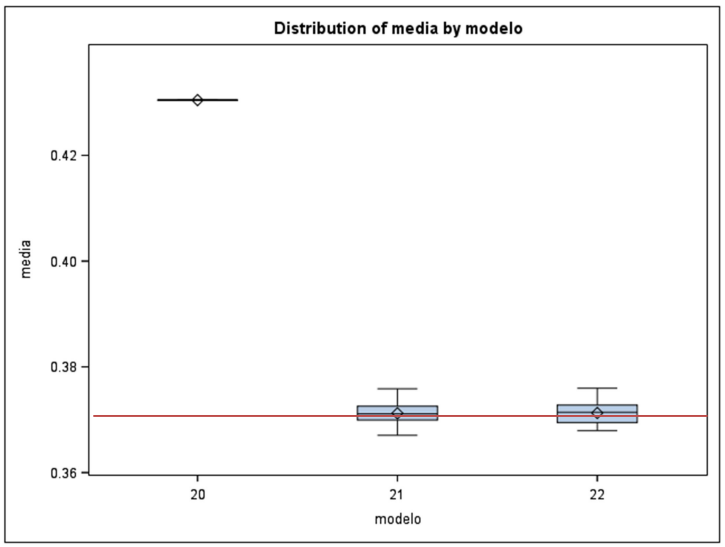
Conjunto Miner: el mejor modelo es el 12.



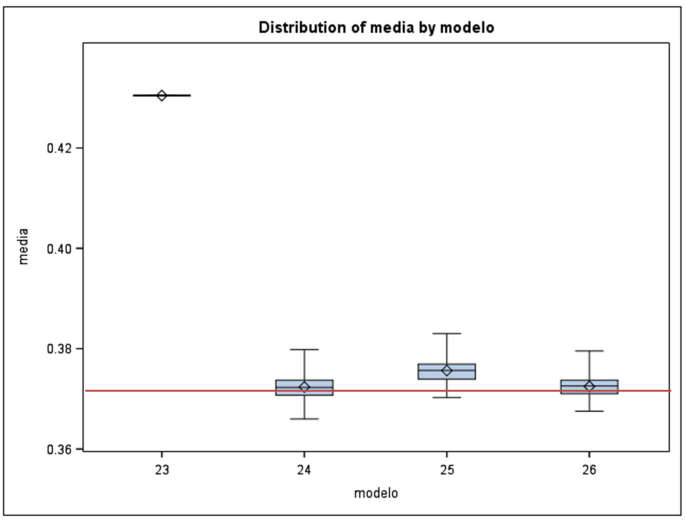
Conjunto importancia: El mejor modelo es el 17.



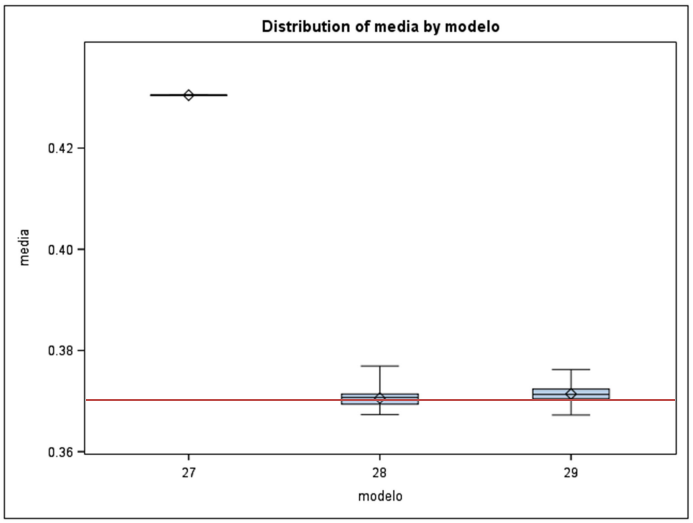
Conjunto Aleatoria 1: El mejor modelo es el 21.



Conjunto Aleatoria 2: el mejor modelo es el número 24.

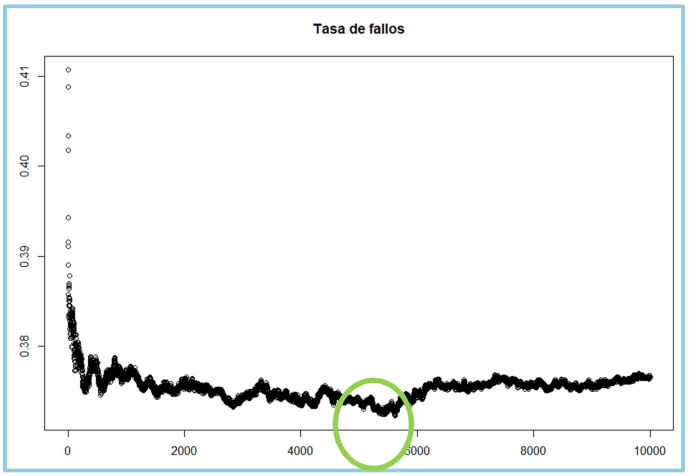


Conjunto mejor con 10: el mejor modelo es el 29.

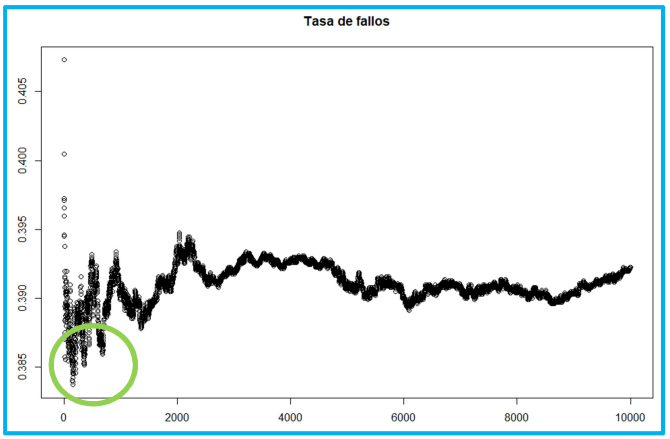


Anexo V. Definición número de árboles a utilizar para Bagging en R

Conjunto Selección Miner: La mínima tasa de fallos se obtiene utilizando entre 5000 y 5500 árboles.

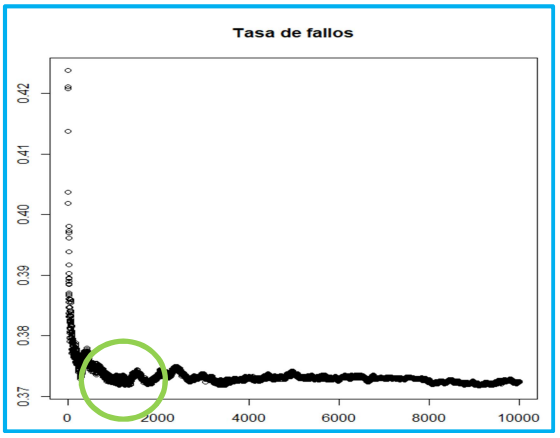


Conjunto Importancia de la variable: Como se puede observar en la gráfica al incrementar el número de árboles, incrementa el error. El error mínimo se consigue con pocas iteraciones se utilizarán 500 árboles.

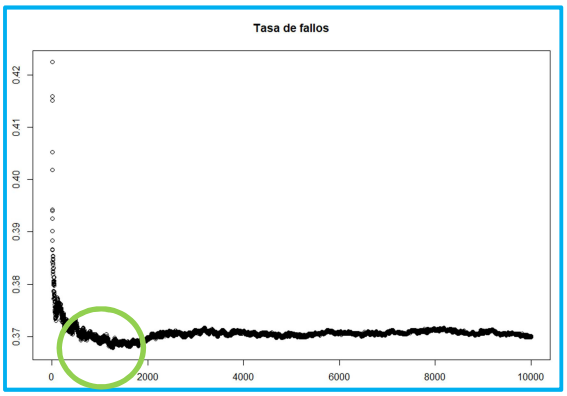


Conjunto Aleatorio 1

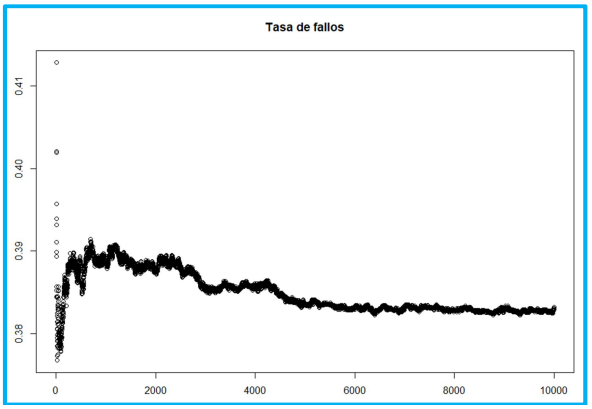
La tasa mínima tasa de fallos se obtiene utilizando cerca de 1000 árboles y después de la 8000. Se probará con 1000.



Conjunto Aleatorio 2: La tasa mínima tasa de fallos se obtiene con 1000 árboles, después de este punto se incrementa el error.

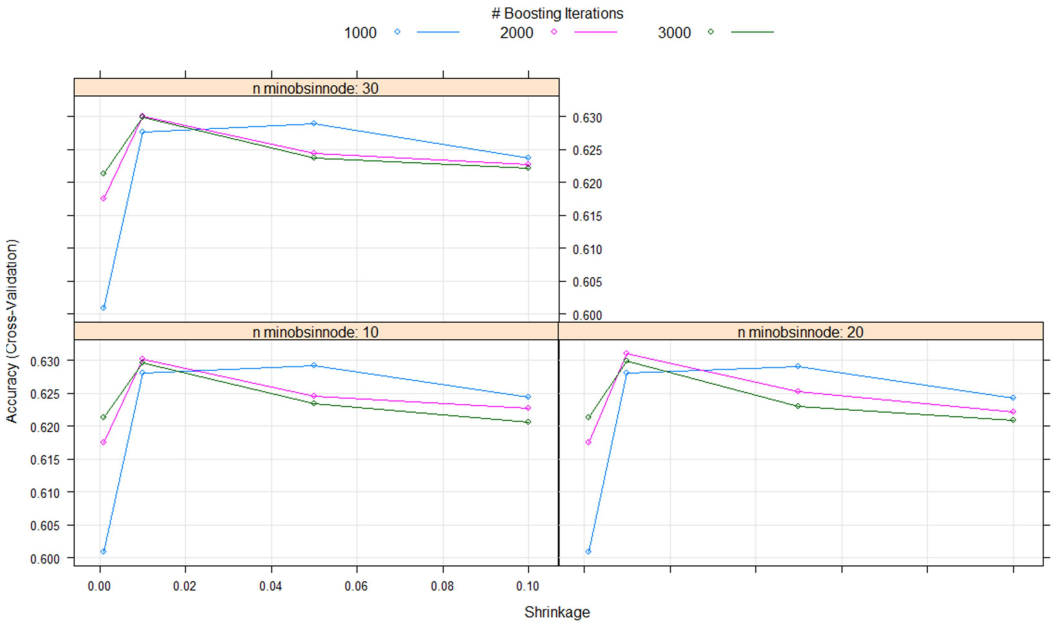


Conjunto Mejor con 10: Con pocos árboles presenta alta variabilidad en los resultados, se estabiliza cerca a los 6000 árboles, se utilizará este valor para las pruebas.

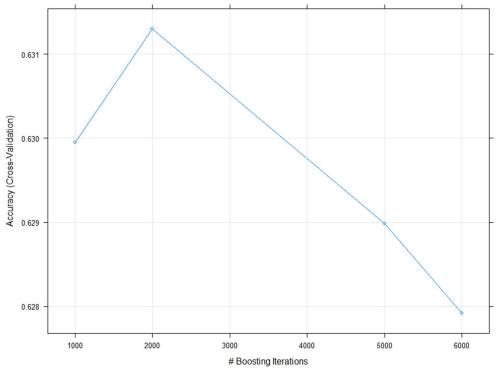


Anexo VI. Definición parámetros incremento gradiente con Caret y pruebas de parada anticipada.

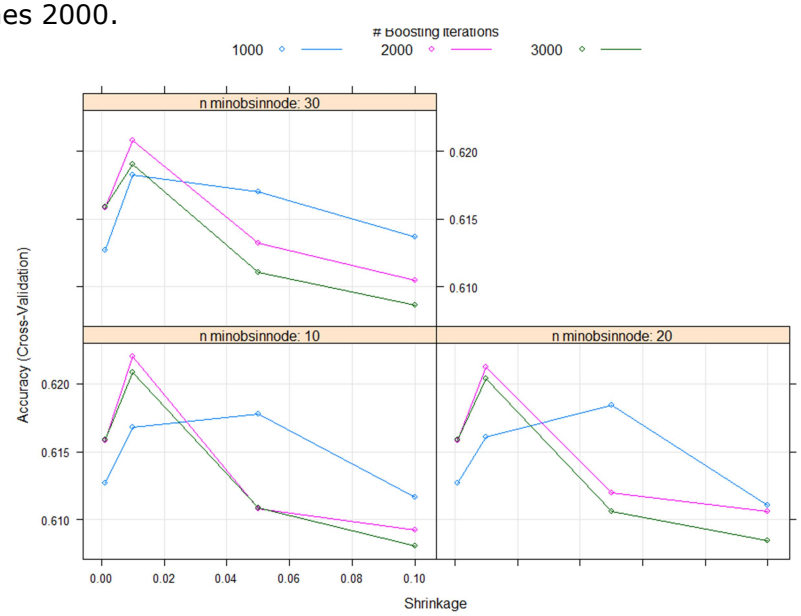
Conjunto Miner: El valor de los parámetros que optimiza la tasa de aciertos es: shrinkage de 0.01, mínimo de observaciones por nodo de 20 y número de iteraciones 2000.



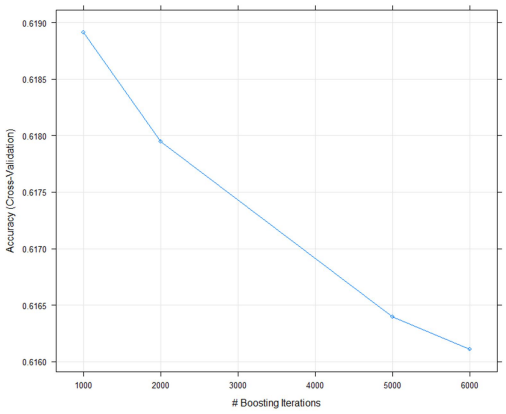
Parada anticipada: Confirma que el número óptimo de iteraciones es 2000, a partir de este número se disminuye la tasa de aciertos al incrementar número de iteraciones, de igual forma al utilizar menos iteraciones la tasa de aciertos es menor.



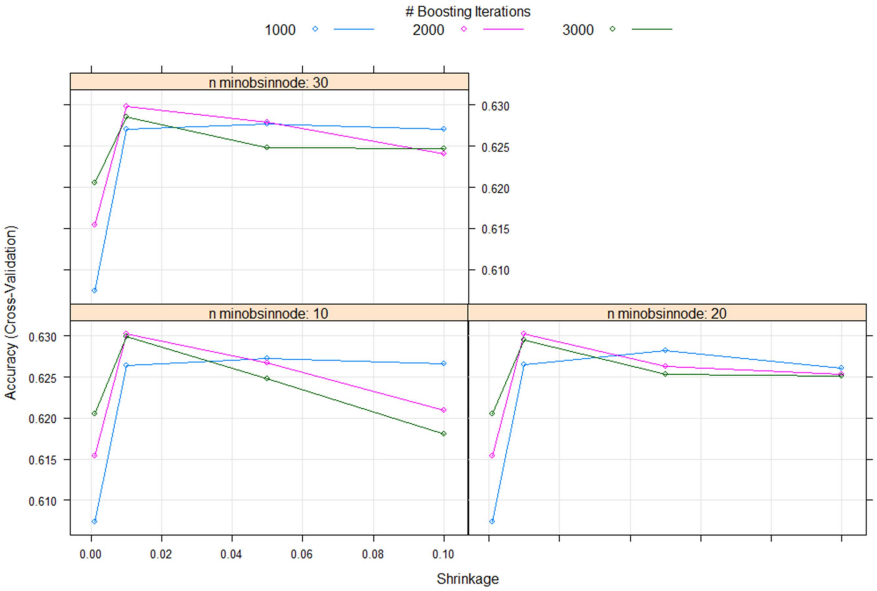
Conjunto Importancia: El valor de los parámetros que optimiza la tasa de aciertos es: shrinkage de 0.01, mínimo de observaciones por nodo de 10 y número de iteraciones 2000.



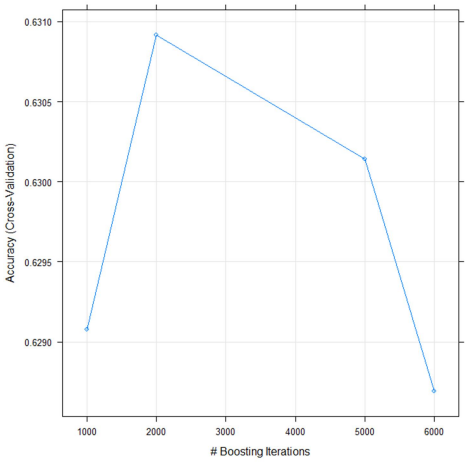
Parada anticipada: Se utilizará 1000 como máximo número de iteraciones, al incrementar las iteraciones se disminuye la tasa de aciertos.



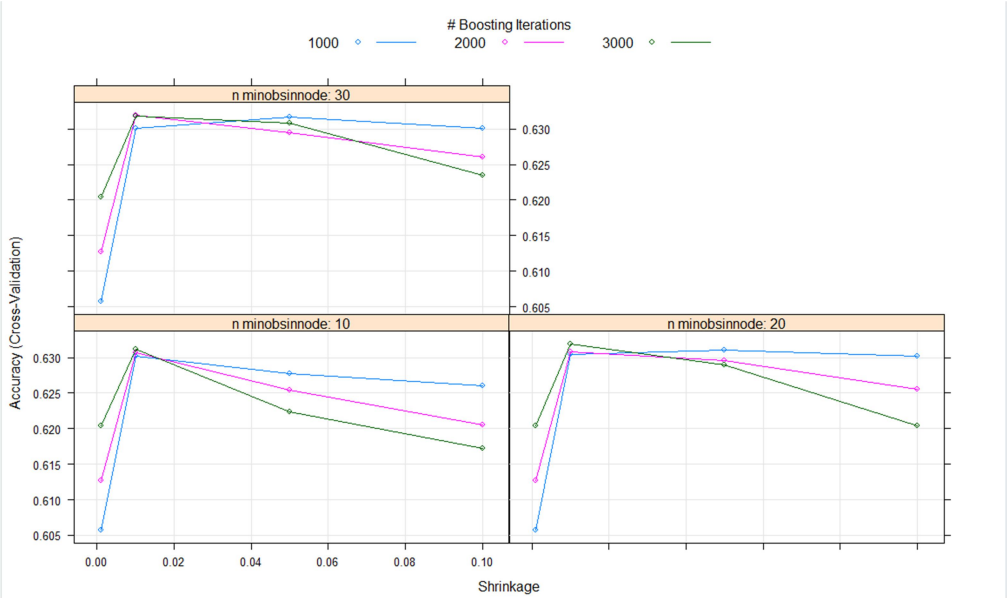
Conjunto Aleatoria 1: El valor de los parámetros que optimiza la tasa de aciertos es: shrinkage de 0.01, mínimo de observaciones por nodo de 20 y número de iteraciones 2000.



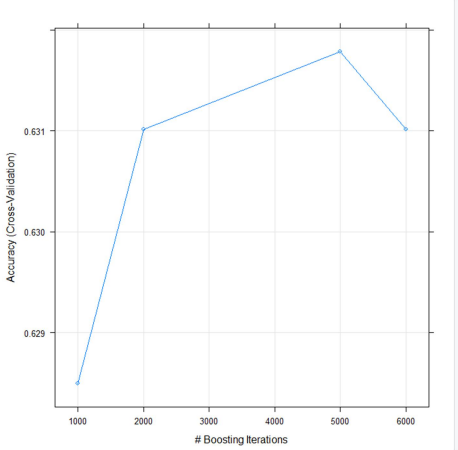
Parada anticipada: Se confirma que el número de iteraciones óptimo es 2000, para valores superiores o inferiores disminuye la tasa de aciertos.



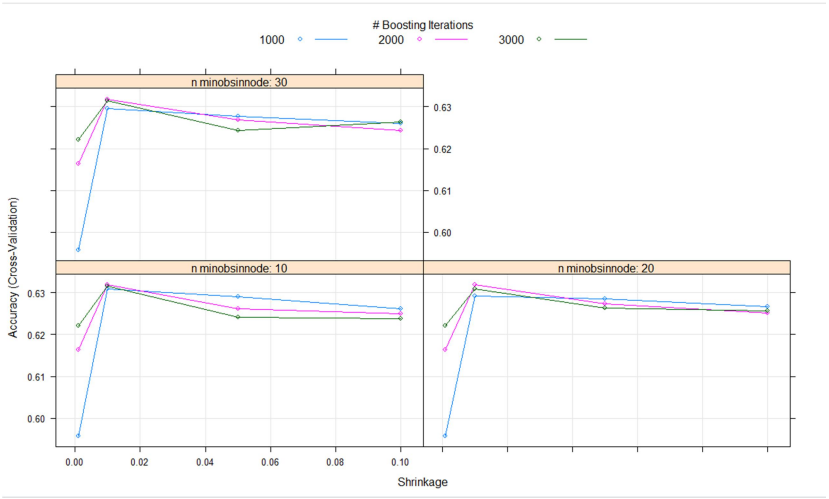
Conjunto Aleatoria 2: El valor de los parámetros que optimiza la tasa de aciertos es: shrinkage de 0.01, mínimo de observaciones por nodo de 30 y número de iteraciones 2000.



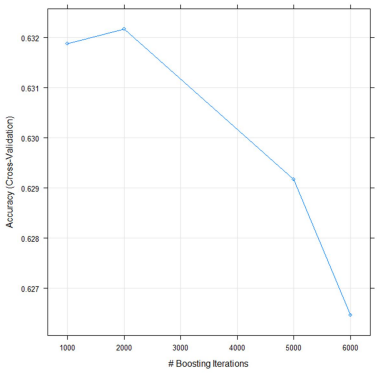
Parada anticipada: Con 5000 iteraciones se consigue una mejor tasa de aciertos por lo cual se probará con este valor.



Conjunto Mejor con 10: El valor de los parámetros que optimiza la tasa de aciertos es: shrinkage de 0.01, mínimo de observaciones por nodo de 10 y número de iteraciones 2000.

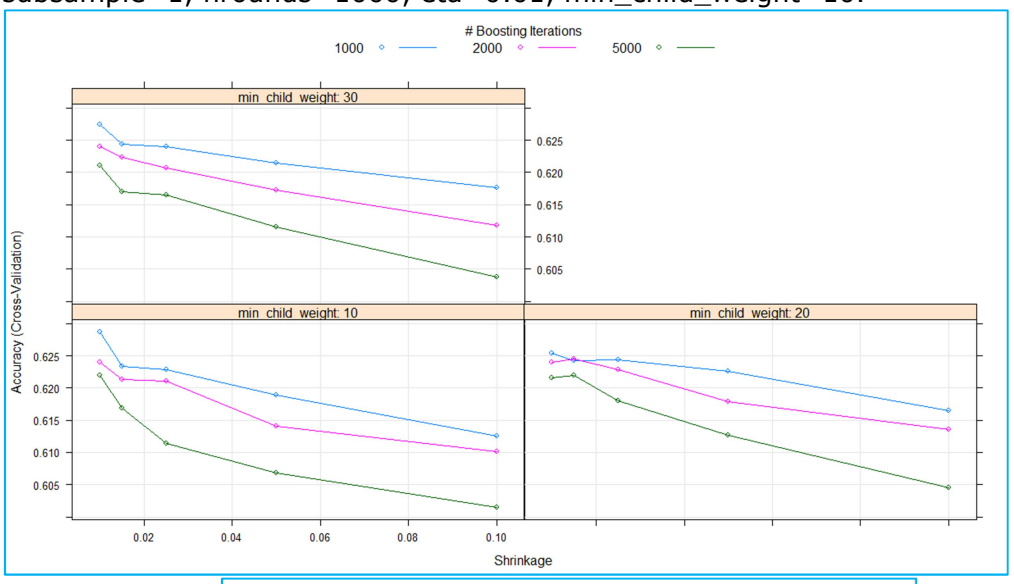


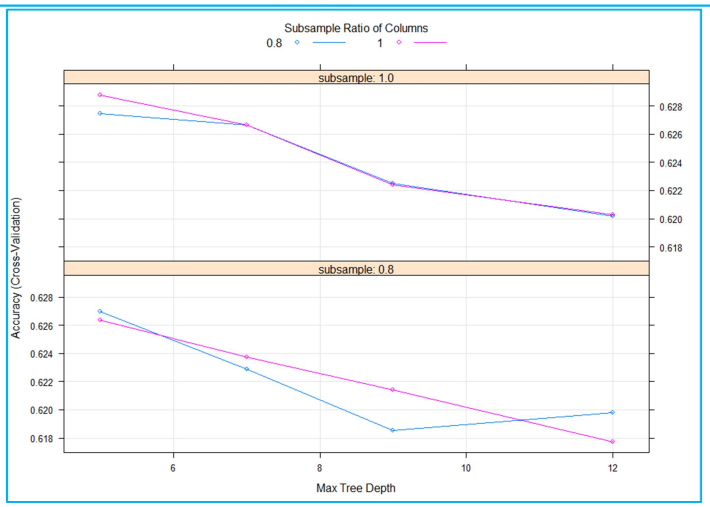
Parada anticipada: Requiere parada anticipada en 2000.



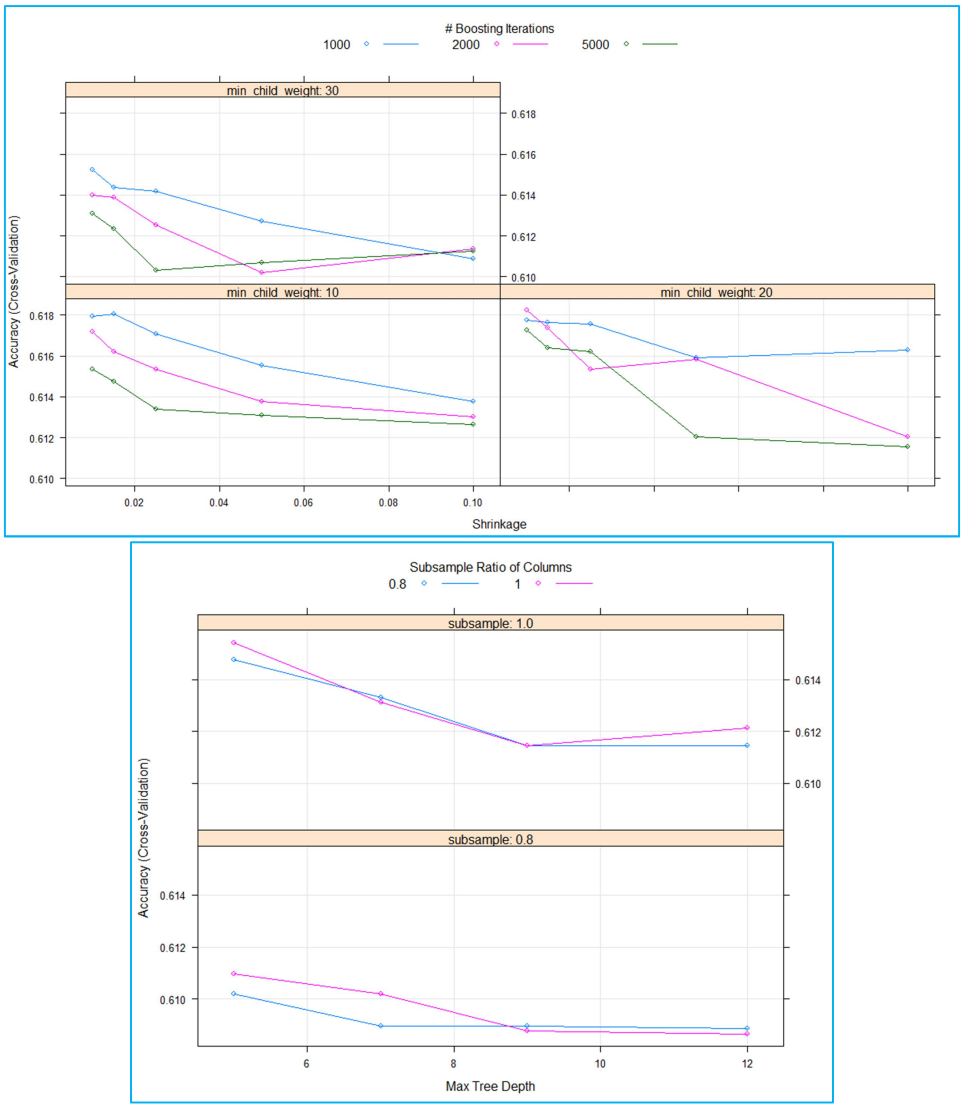
Anexo VII. Definición parámetros Xgboost con Caret

Conjunto Miner: Los parámetros que maximizan la tasa de aciertos para este conjunto de variables son: max_depth=5, gamma=0, colsample_bytree=1, subsample=1, nrounds=1000, eta=0.01, min_child_weight=10.

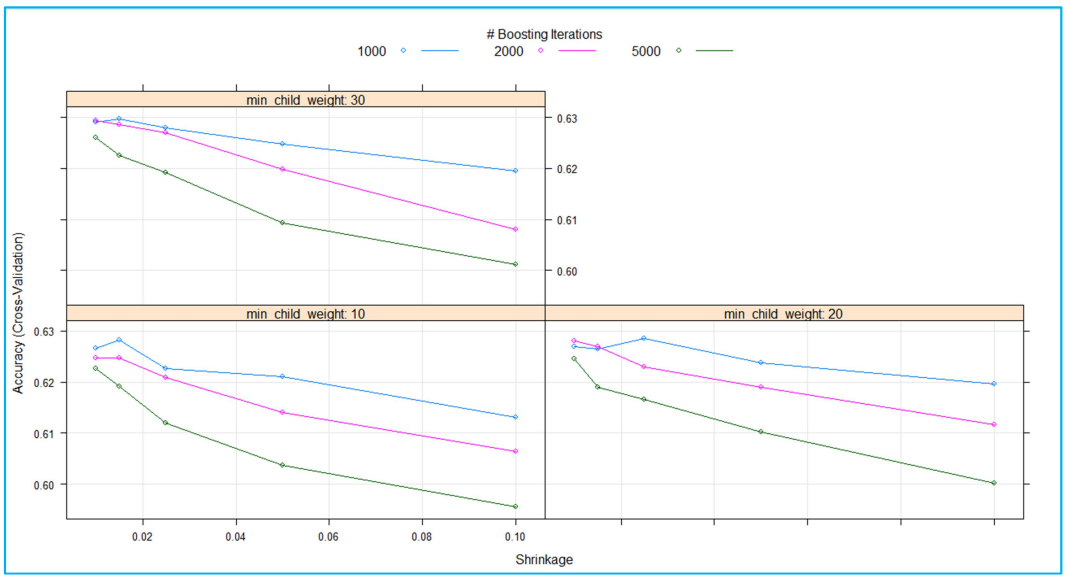


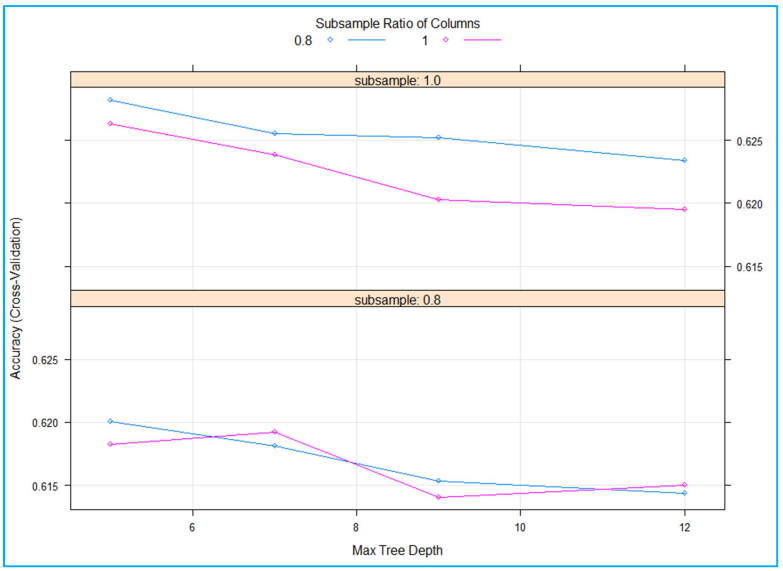


Importancia: Los parámetros que maximizan la tasa de aciertos para este conjunto de variables son: max_depth=5, gamma=0, colsample_bytree=1, subsample=1 son: nrounds=2000, eta=0.01, min_child_weight=20.

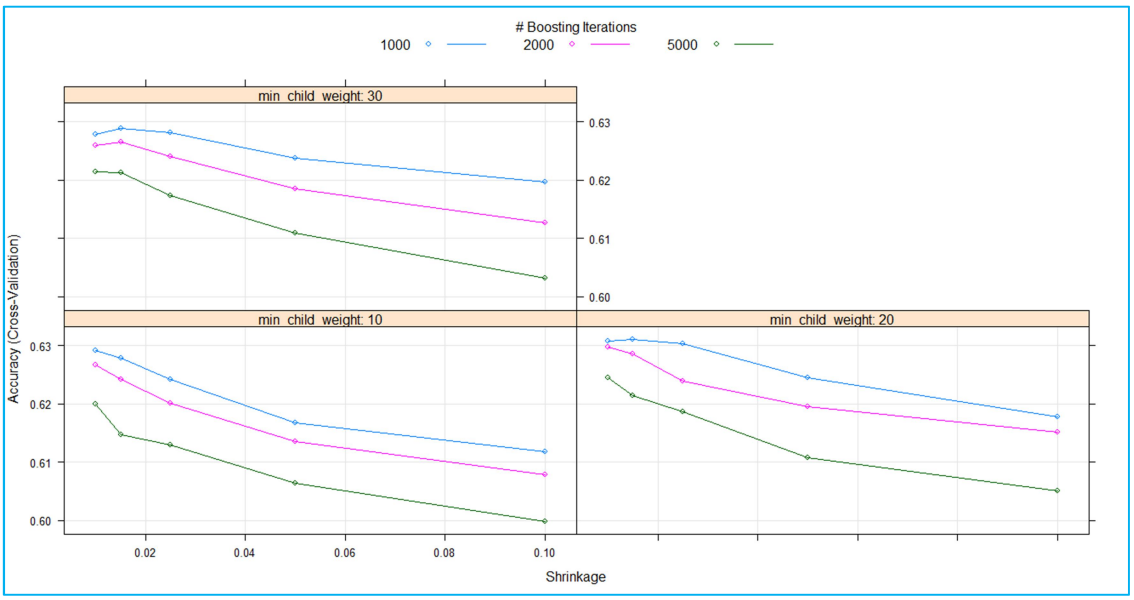


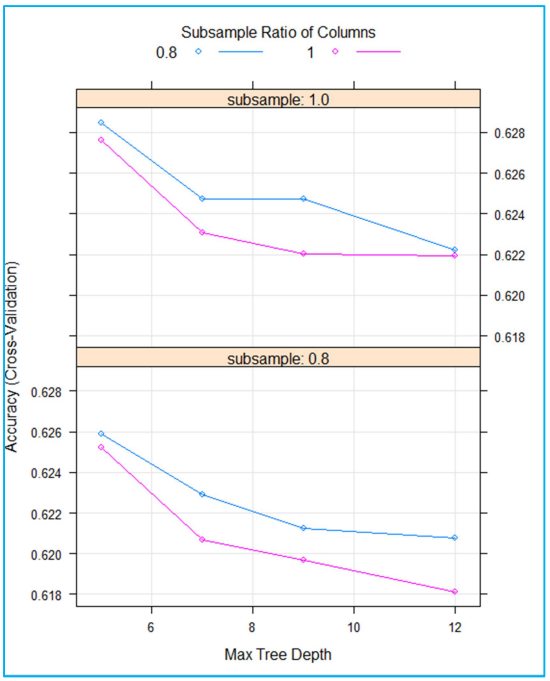
Aleatoria 1: Los parámetros que maximizan la tasa de aciertos para este conjunto de variables son: max_depth=5, gamma=0, colsample_bytree=1, subsample=1, nrounds=1000, eta=0.015, min_child_weight=30.



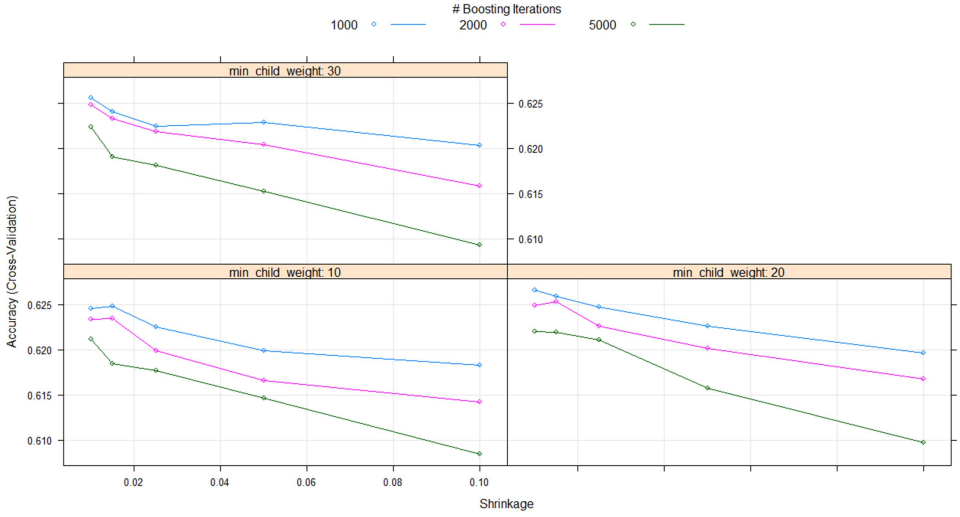


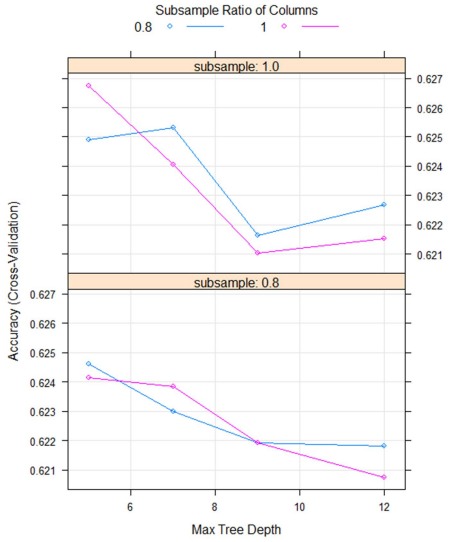
Aleatoria 2: Los parámetros que maximizan la tasa de aciertos para este conjunto de variables dejando fijos: max_depth=5, gamma=0, colsample_bytree=1, subsample=1 son: nrounds=1000, eta=0.015, min_child_weight=20.





Mejor con 10: Los parámetros que maximizan la tasa de aciertos para este conjunto de variables son: max_depth=5, gamma=0, colsample_bytree=1, subsample=1, nrounds=1000, eta=0.01, min_child_weight=20.

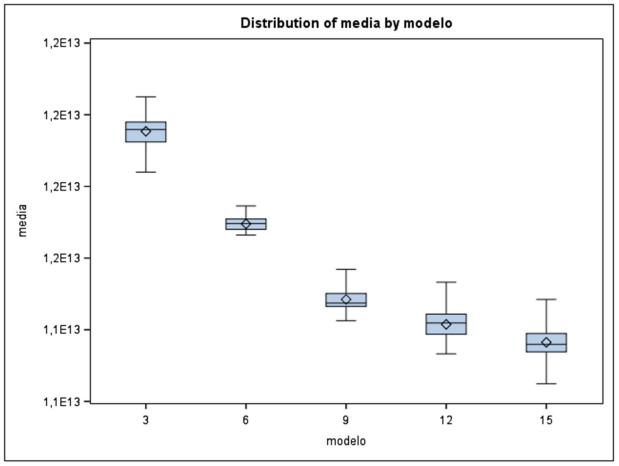




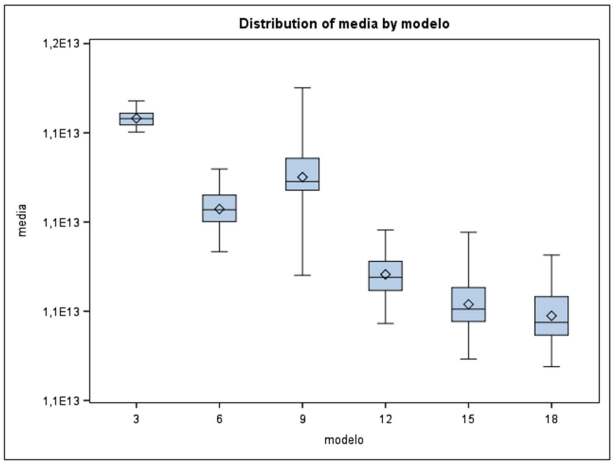
ANEXOS PARTE II VARIABLE OBJETIVO CONTINUA

Anexo VIII. Definición número de nodos para redes con SAS Base

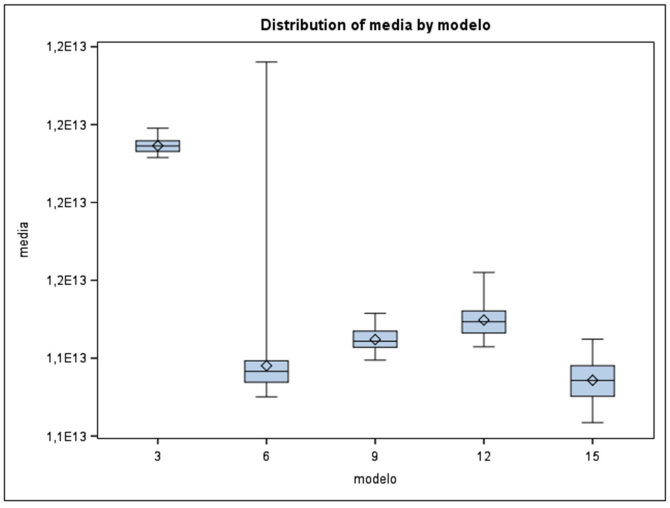
Conjunto Selección Miner: La menor tasa de fallos se consigue utilizando 15 nodos.



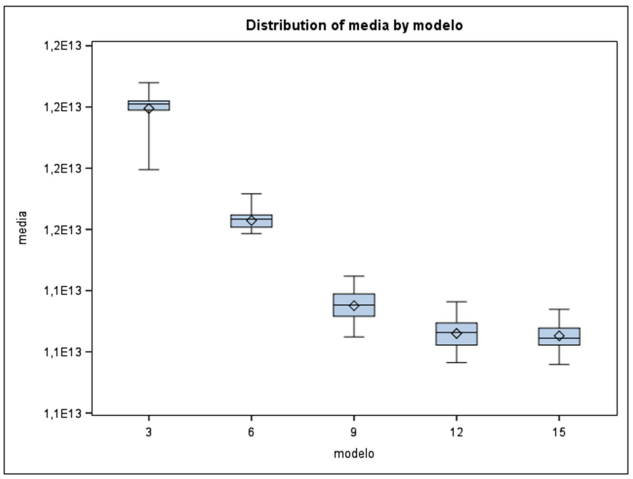
Conjunto importancia: La menor tasa de fallos se consigue utilizando 18 nodos.



Conjunto Aleatoria 1: La menor tasa de fallos se consigue utilizando 15 nodos.

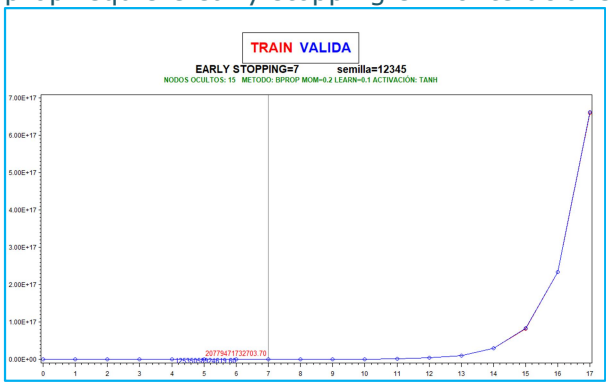


Conjunto Aleatoria 2: La menor tasa de fallos se consigue utilizando 15 nodos.

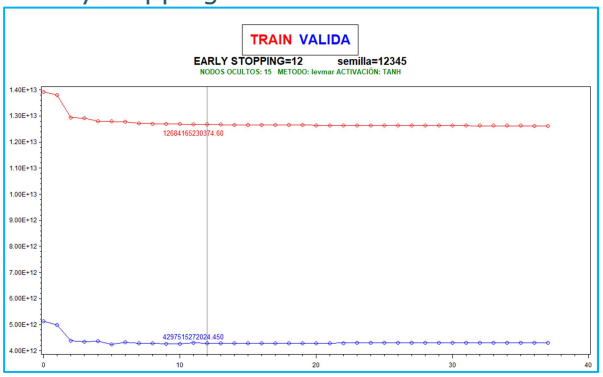


Anexo IX Definición de parada anticipada redes neuronales variable objetivo continua

Conjunto Miner: Bprop requiere early stopping en 10 iteraciones.

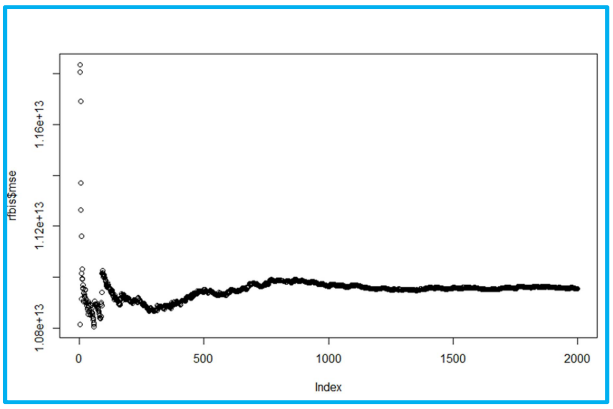


Levmar: no requiere early stopping

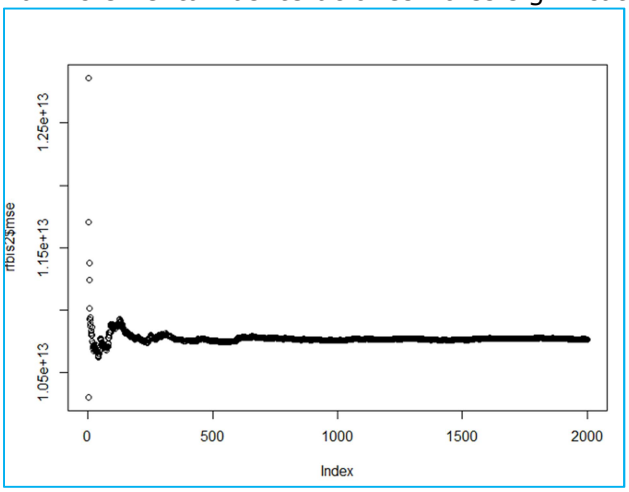


Anexo X. Definición número de árboles a utilizar para Bagging en R

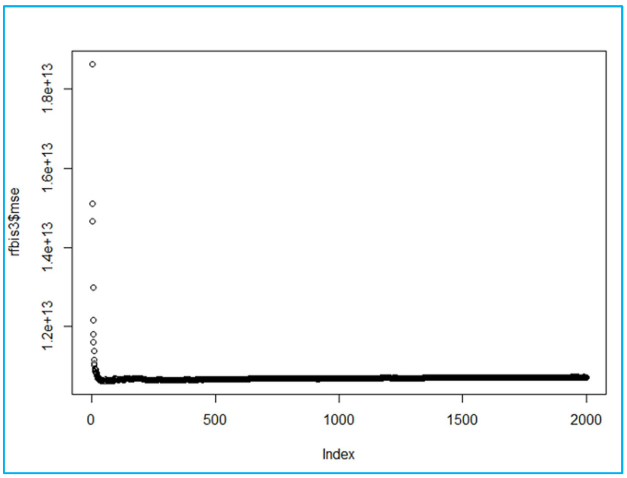
Conjunto Miner: El mínimo error se obtiene con menos de 500 árboles, se probará con 400.



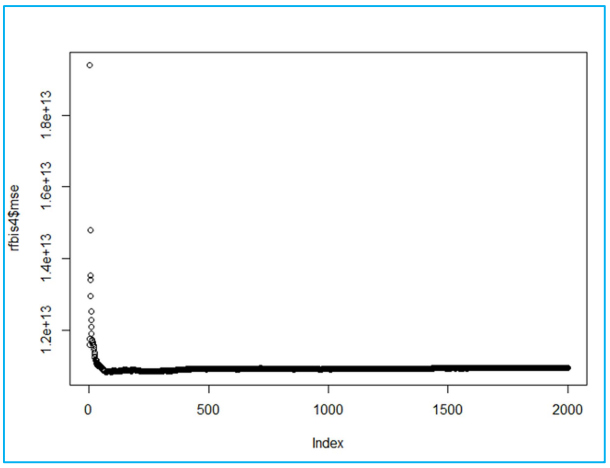
Conjunto importancia: El mínimo error se obtiene cerca a las 500 iteraciones, aunque la variación al incrementar las iteraciones no es significativa.



Conjunto Aleatoria 1: No requiere parada anticipada, se probará con 1000 iteraciones.

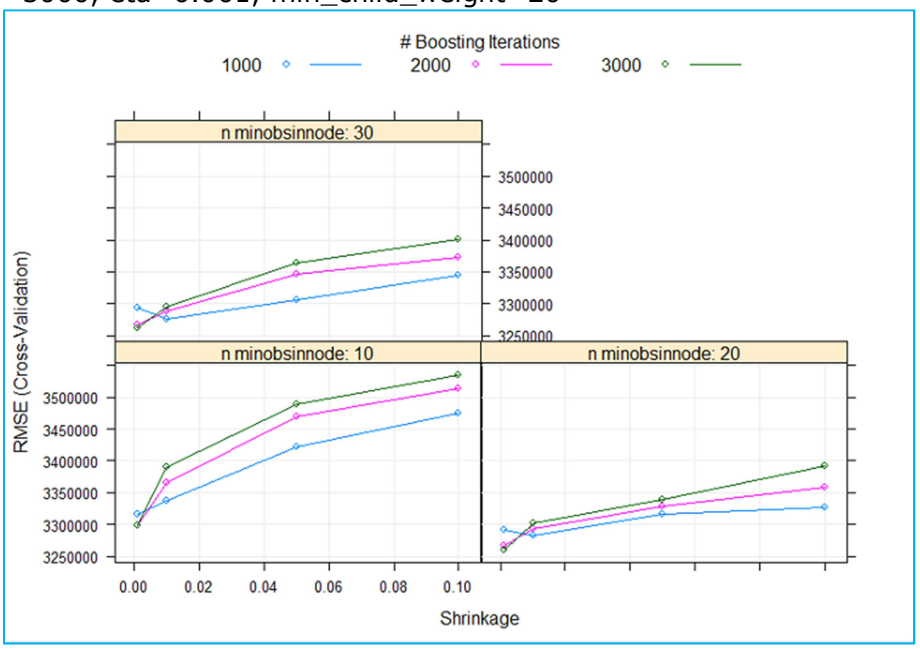


Conjunto Aleatoria 2: El error incrementa un poco después de la iteración 1000, aunque no es significativo.

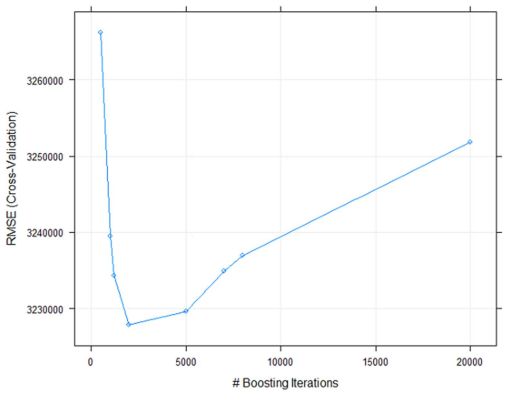


Anexo XI. Definición parámetros incremento gradiente con Caret y pruebas de parada anticipada.

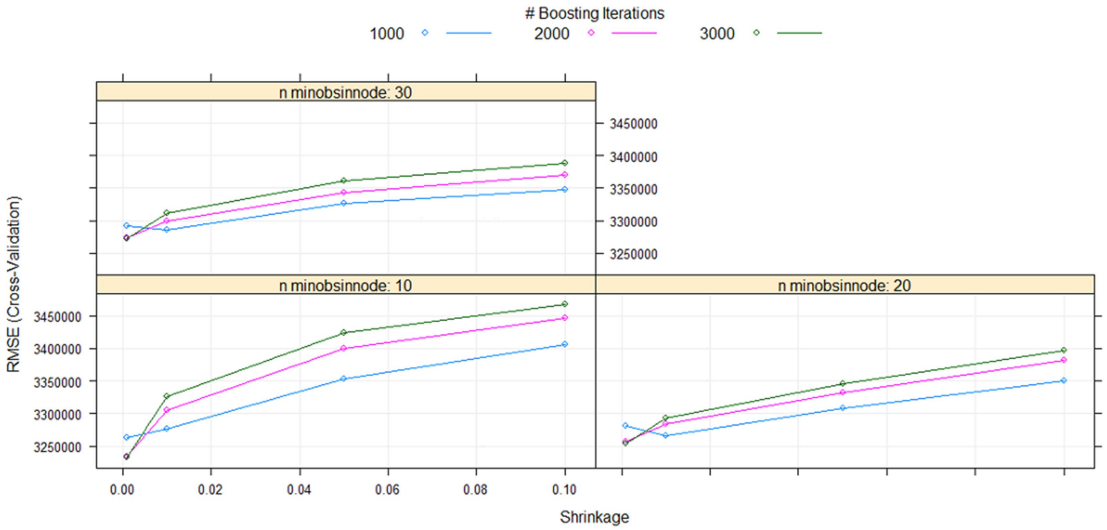
Conjunto Miner: Los parámetros que minimizan el error para este conjunto de variables son: max_depth=5, gamma=0, colsample_bytree=1, subsample=1, nrounds=3000, eta=0.001, min_child_weight=20



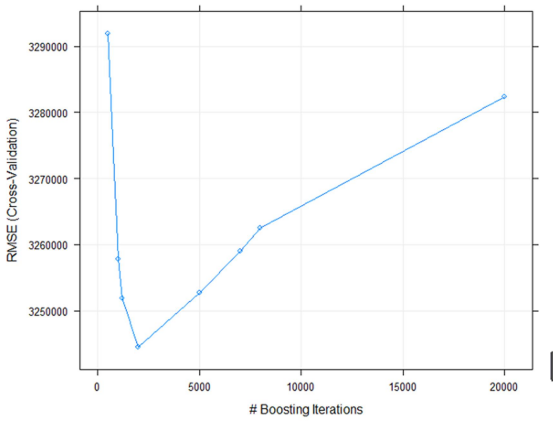
Parada anticipada: Se debe realizar parada anticipada en 2000 iteraciones.



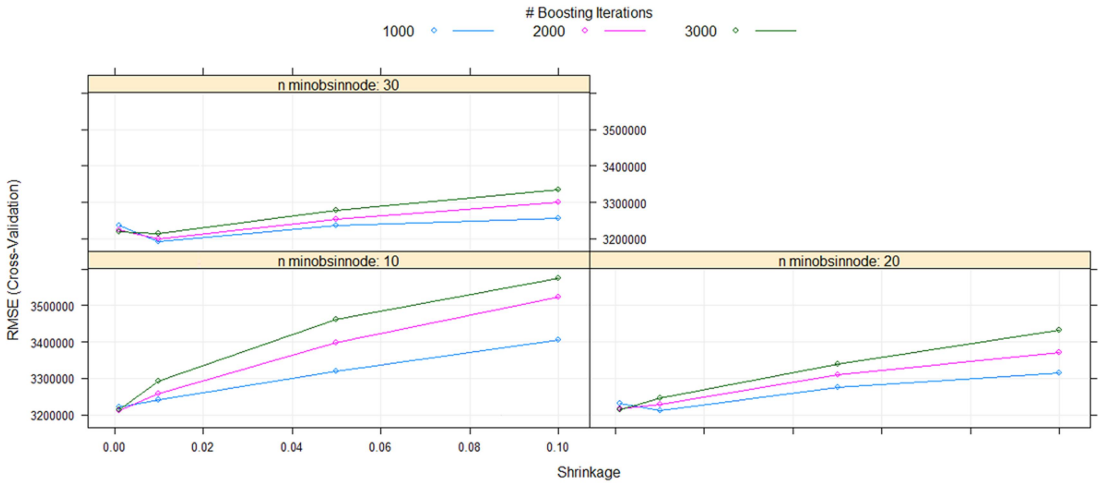
Conjunto Importancia: Los parámetros que minimizan el error para este conjunto de variables son: max_depth=5, gamma=0, colsample_bytree=1, subsample=1, nrounds=3000, eta=0.001, min_child_weight=10



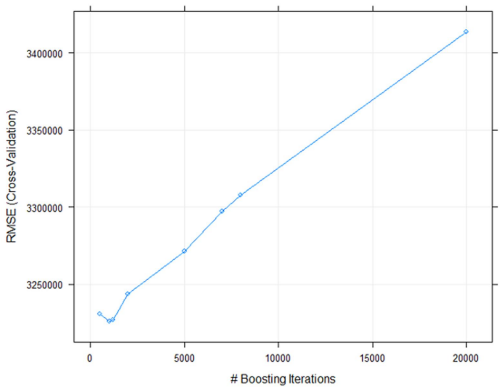
Parada anticipada: Se debe realizar parada anticipada en 2000 iteraciones.



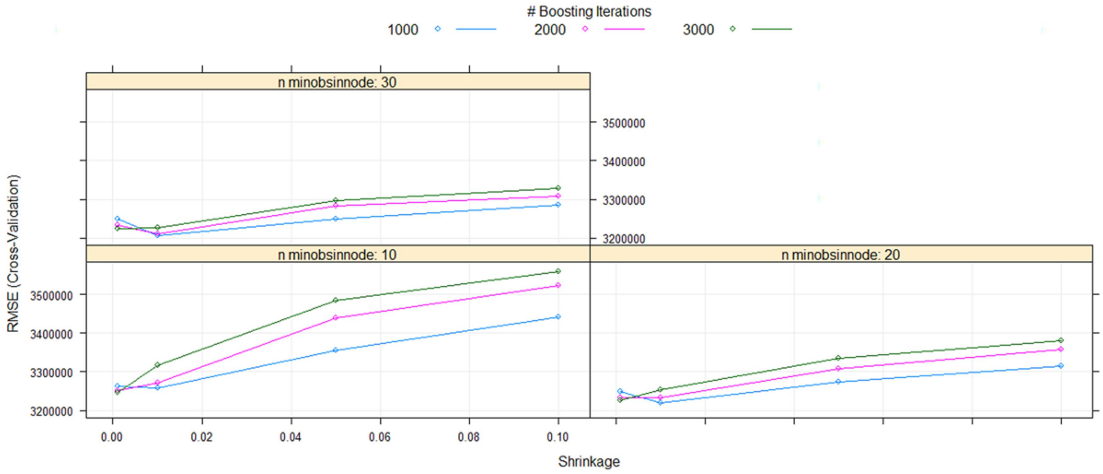
Conjunto Aleatoria 1: Los parámetros que minimizan el error para este conjunto de variables son: max_depth=5, gamma=0, colsample_bytree=1, subsample=1, nrounds=1000, eta=0.01, min_child_weight=30



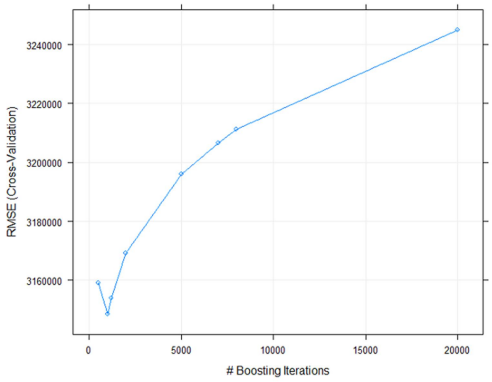
Parada anticipada: Se comprueba que el valor óptimo de iteraciones es 1000.



Conjunto Aleatoria 2: Los parámetros que minimizan el error para este conjunto de variables son: max_depth=5, gamma=0, colsample_bytree=1, subsample=1, nrounds=1000, eta=0.01, min_child_weight=30

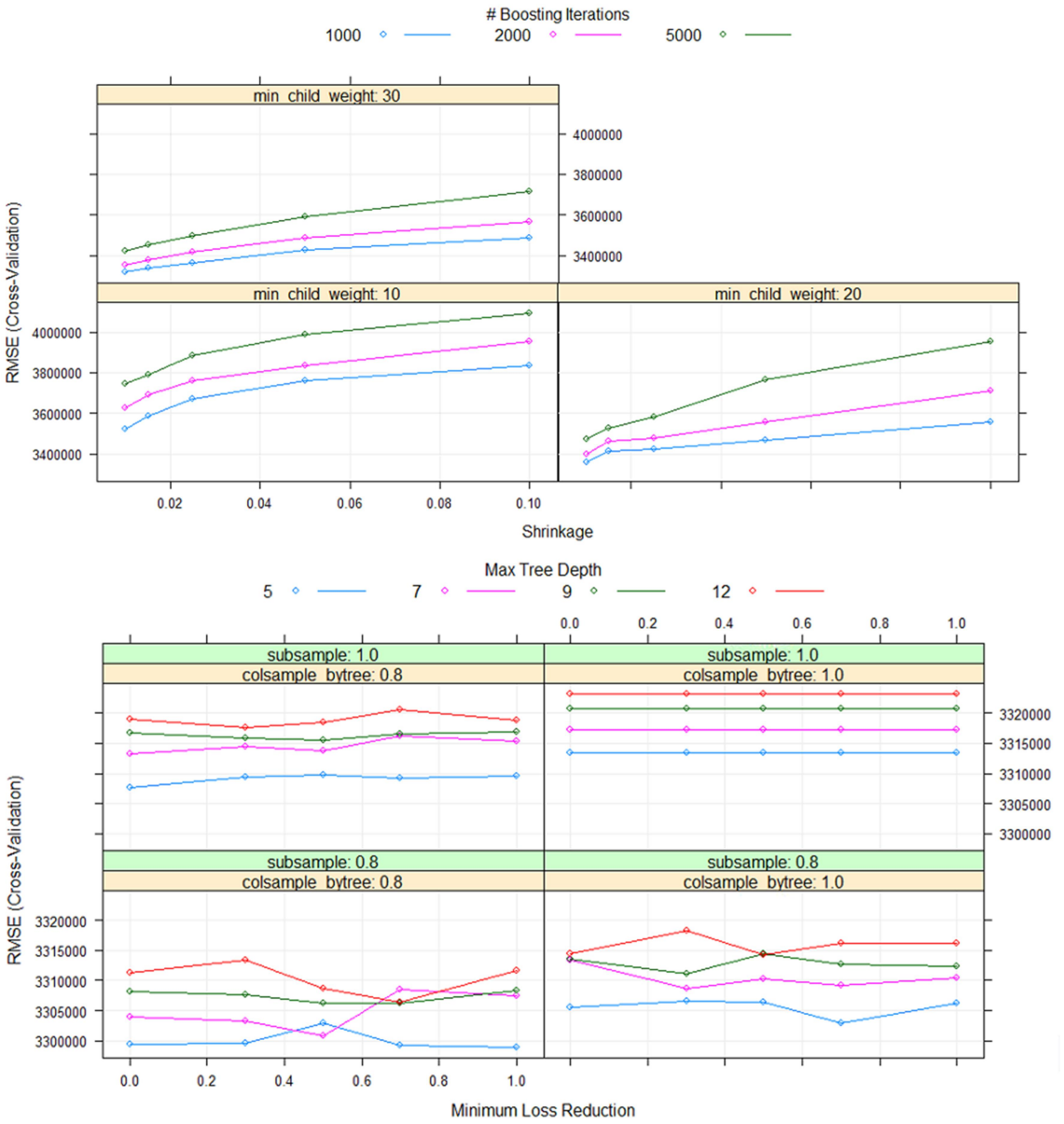


Parada anticipada: Se comprueba que el valor óptimo de iteraciones es 1000.

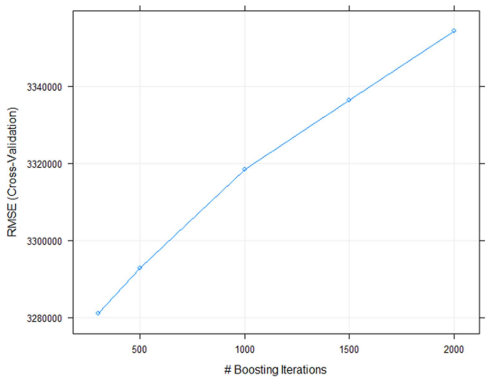


Anexo XII. Definición parámetros Xgboost con Caret

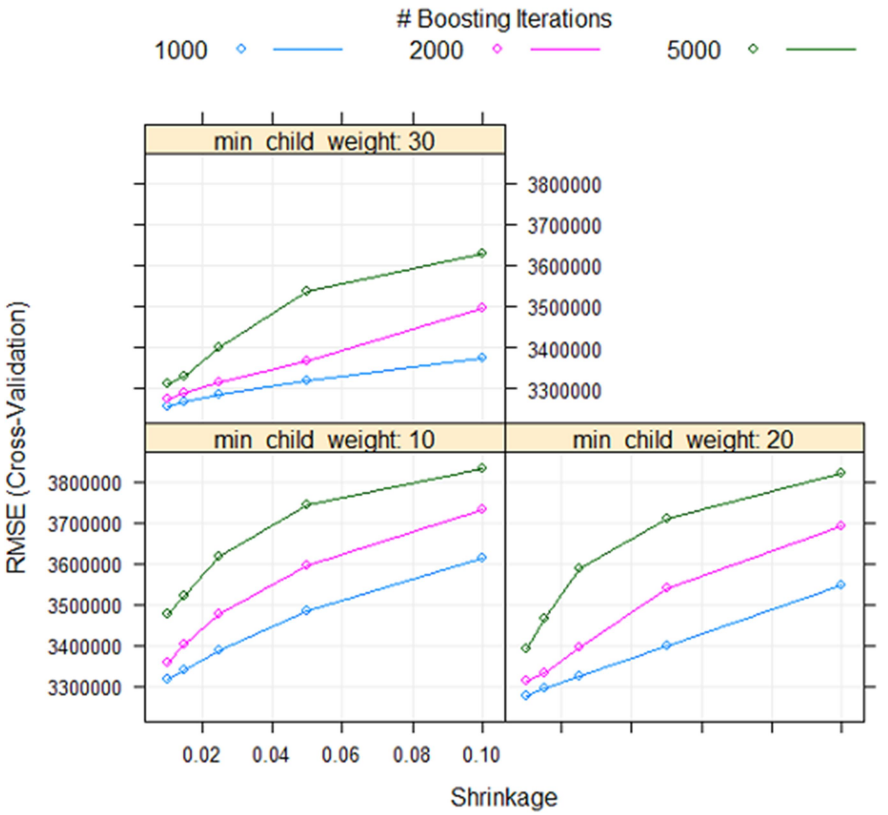
Conjunto Miner: Los parámetros que minimizan el error para este conjunto de variables son: max_depth=5, gamma=0, colsample_bytree=1, subsample=1, nrounds=1000, eta=0.01, min_child_weight=30

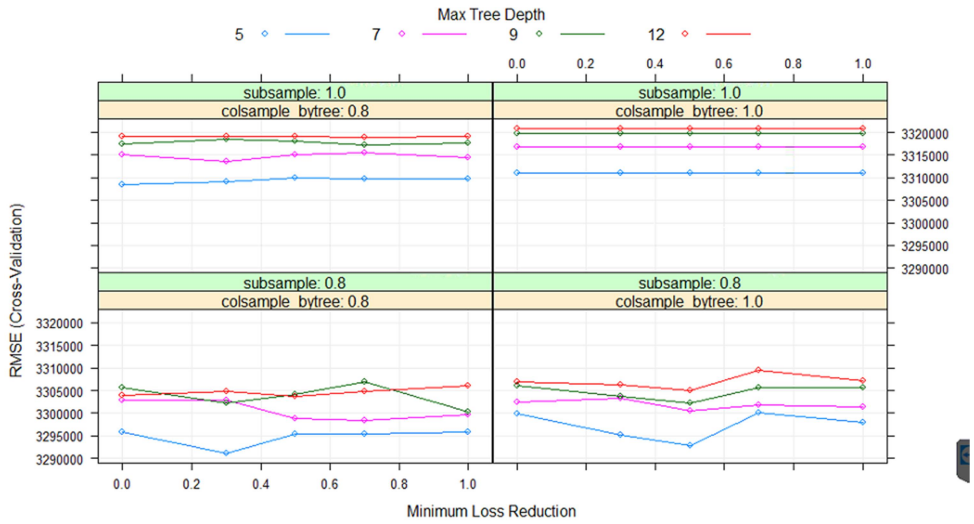


Parada anticipada: Requiere parada anticipada en 300 iteraciones.

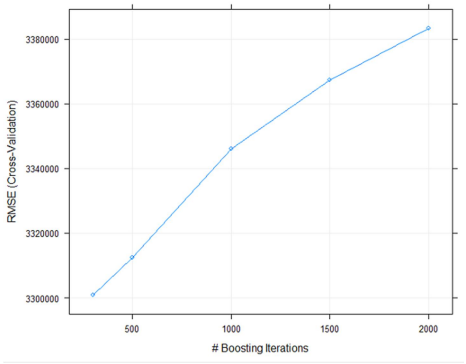


Conjunto Importancia: Los parámetros que minimizan el error para este conjunto de variables son: max_depth=5, gamma=0.3, colsample_bytree=0.8, subsample=0.8, nrounds=1000, eta=0.01, min_child_weight=30

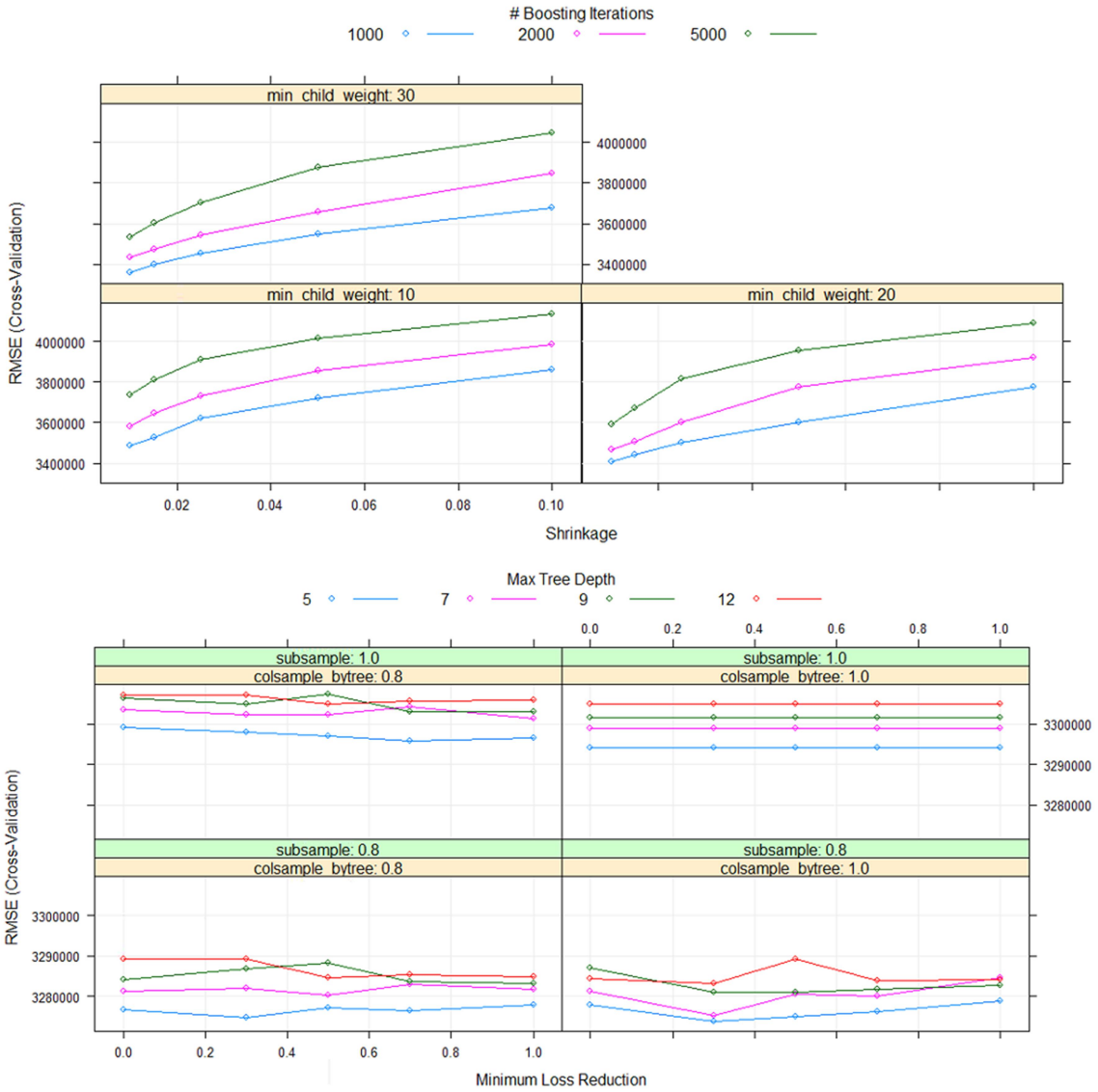




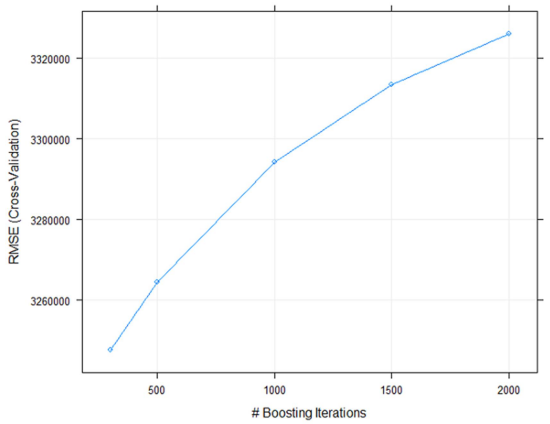
Parada anticipada: Requiere parada anticipada en 300 iteraciones.



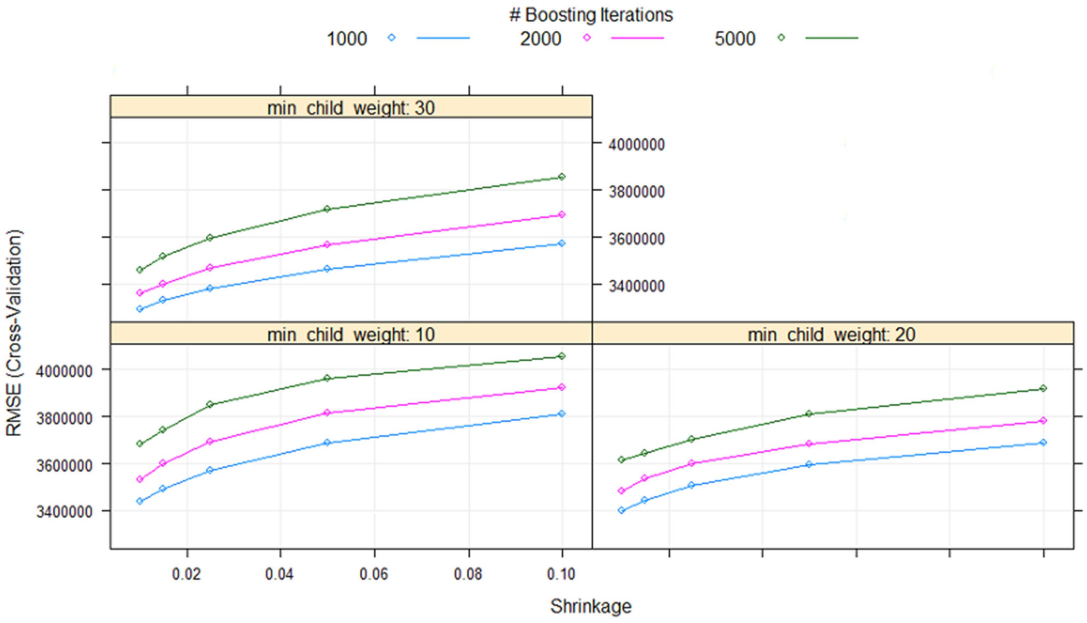
Conjunto Aleatoria 1: Los parámetros que minimizan el error para este conjunto de variables son: max_depth=5, gamma=0.3, colsample_bytree=1, subsample=0.8, nrounds=1000, eta=0.01, min_child_weight=30

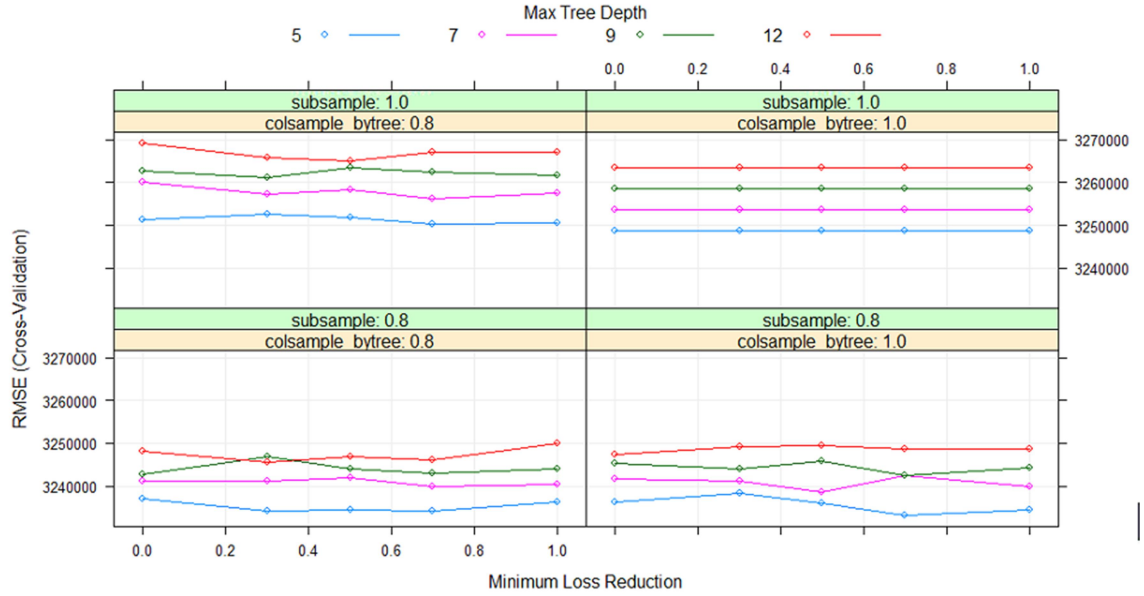


Parada anticipada: Requiere parada anticipada en 300 iteraciones.

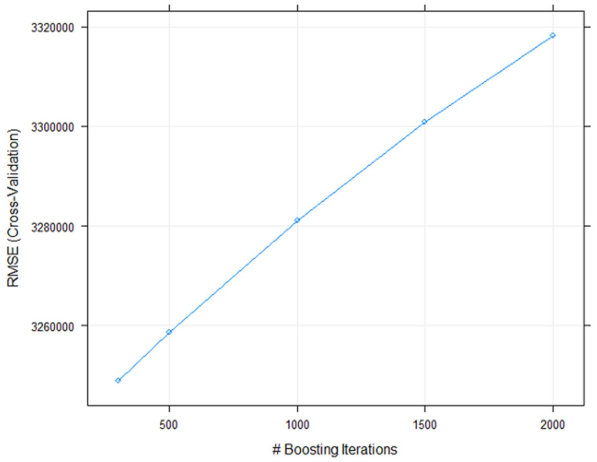


Conjunto Aleatoria 2: Los parámetros que minimizan el error para este conjunto de variables son: max_depth=5, gamma=0.7, colsample_bytree=1, subsample=0.8, nrounds=1000, eta=0.01, min_child_weight=30





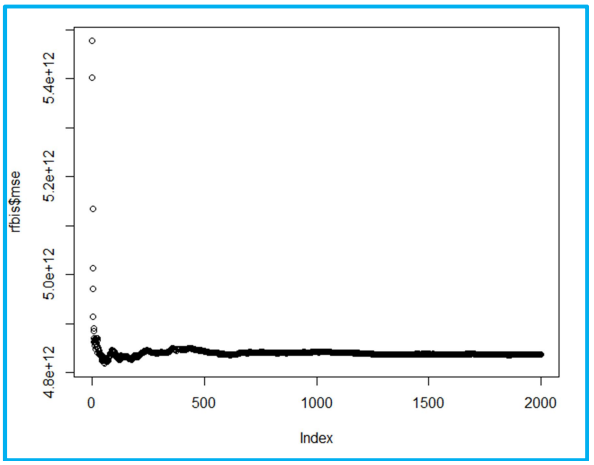
Parada anticipada: Requiere parada anticipada en 300 iteraciones.



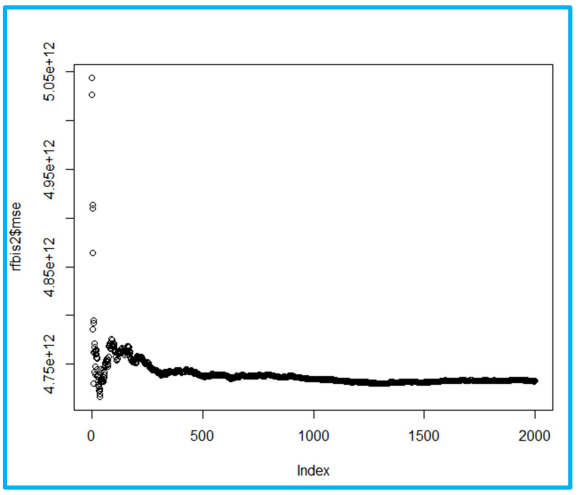
ANEXOS PREDICCIÓN DEL COSTE TOTAL VARIABLE OBJETIVO CONTINUA

Anexo XIII. Definición número de árboles para Bagging en R

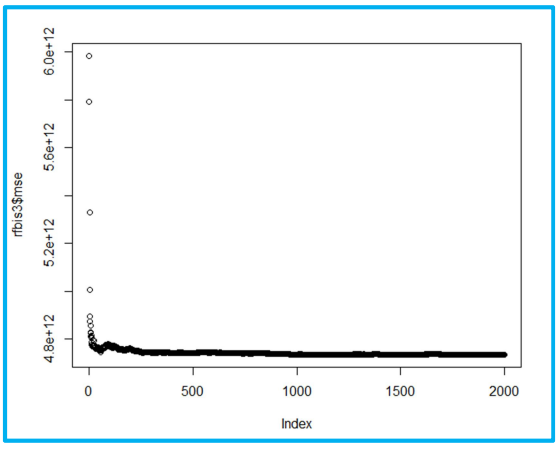
Conjunto Selección Miner: El error mínimo se obtiene utilizando 1500 árboles, a partir de este valor se estabiliza el resultado.



Conjunto importancia: El mínimo error se obtiene utilizando 1200 árboles

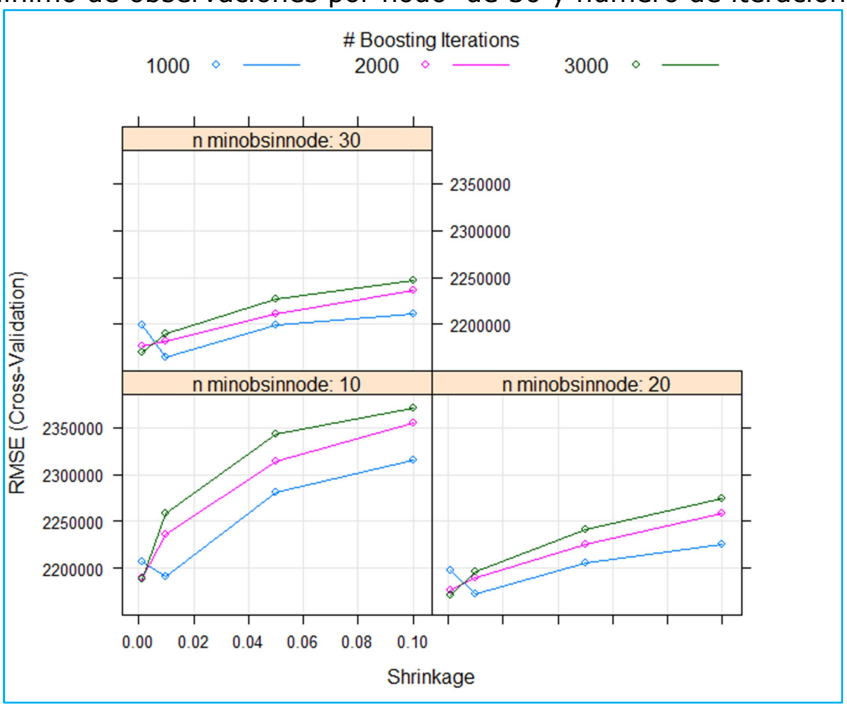


Conjunto Aleatoria: El mínimo error se obtiene utilizando 1000 árboles, a partir de este valor se estabiliza el resultado.

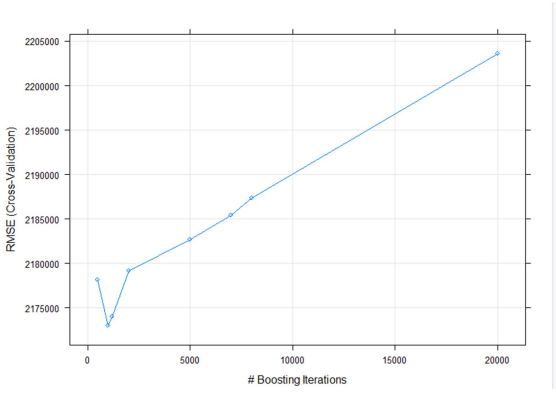


Anexo XIV. Definición parámetros incremento gradiente con Caret y pruebas de parada anticipada.

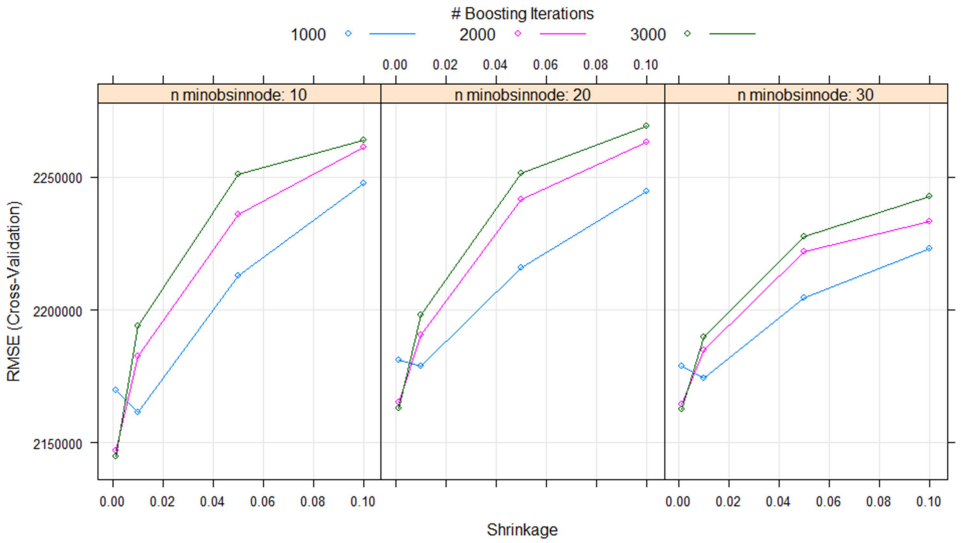
Conjunto Miner: El valor de los parámetros que optimiza el RMSE es: shrinkage de 0.01, mínimo de observaciones por nodo de 30 y número de iteraciones 1000.



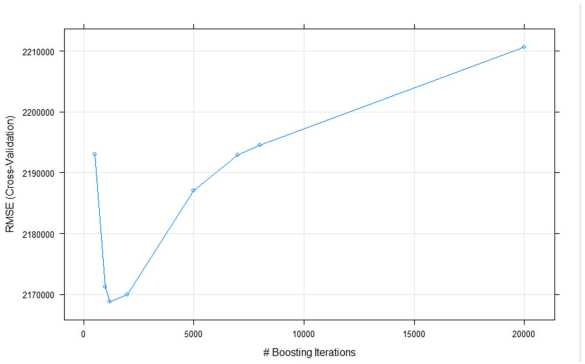
Parada anticipada: Confirma que el número óptimo de iteraciones es 1000, a partir de este número se incrementa el eror al incrementar número de iteraciones.



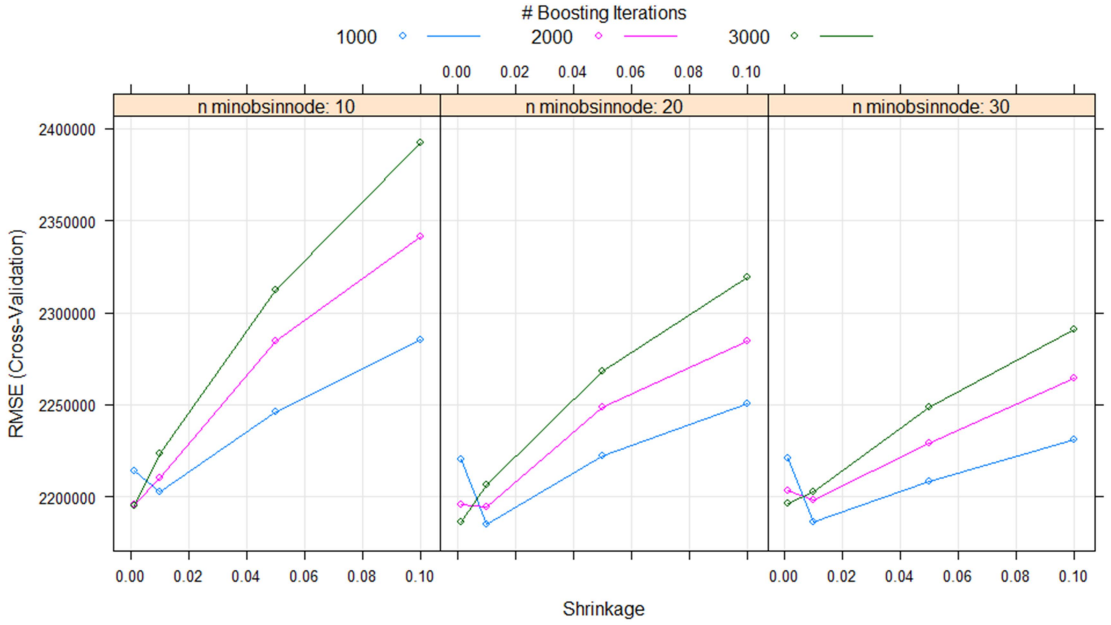
Conjunto importancia: El valor de los parámetros que optimiza el error es: shrinkage de 0.001, mínimo de observaciones por nodo de 10 y número de iteraciones 3000.



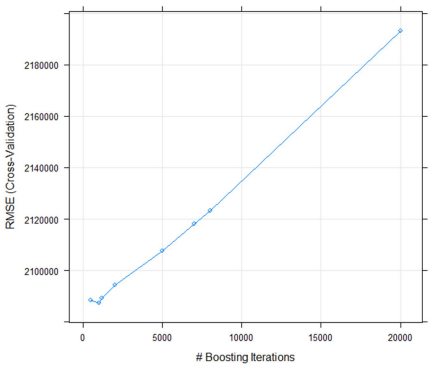
Parada anticipada: Confirma que el número óptimo de iteraciones es 1200, a partir de este número se incrementa el error al incrementar número de iteraciones.



Conjunto aleatoria: El valor de los parámetros que optimiza el RMSE es: shrinkage de 0.01, mínimo de observaciones por nodo de 20 y número de iteraciones 1000

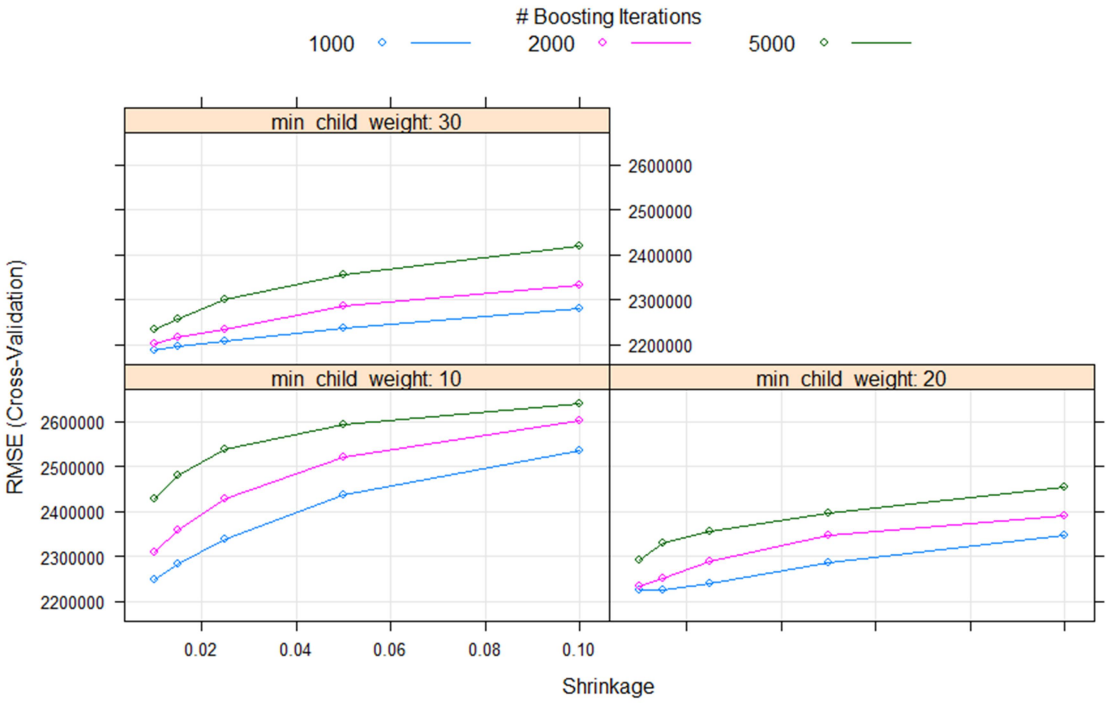


Parada anticipada: Confirma que el número óptimo de iteraciones es 1000, a partir de este número se incrementa el error al incrementar número de iteraciones.

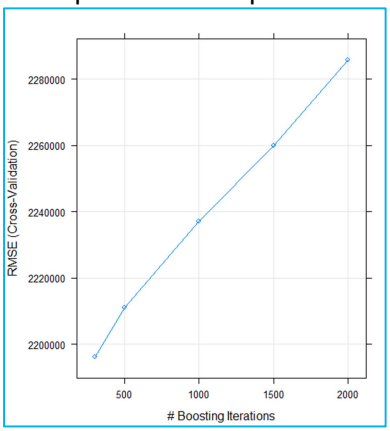


Anexo XV. Definición parámetros Xgboost con Caret

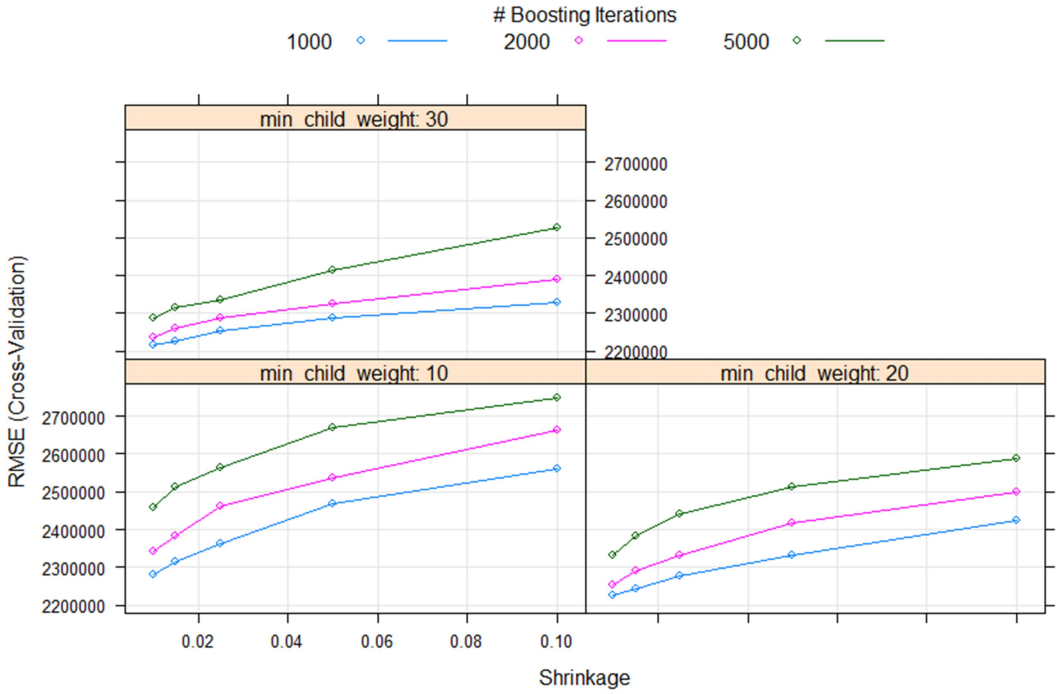
Conjunto Miner: Los parámetros que minimizan el error para este conjunto de variables son: max_depth=5, gamma=0, colsample_bytree=1, subsample=1, nrounds=1000, eta=0.05, min_child_weight=30



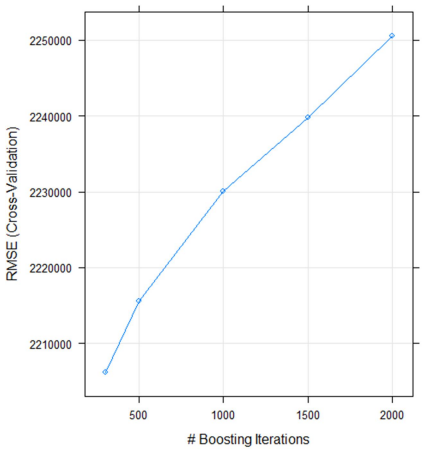
Parada anticipada: Requiere parada anticipada en la iteración 300.



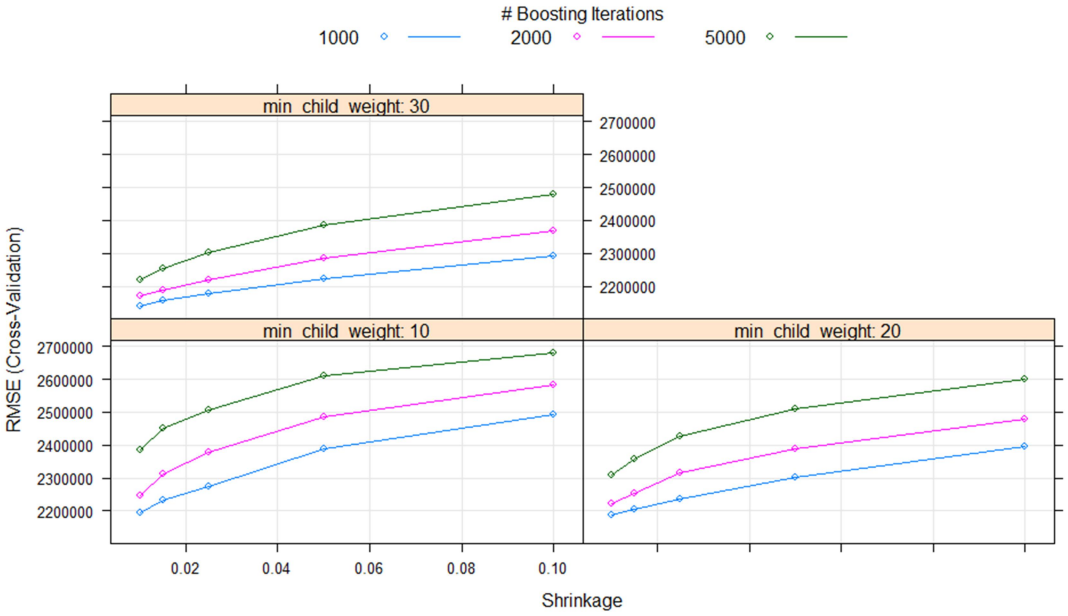
Conjunto Importancia: Los parámetros que minimizan el error para este conjunto de variables son: max_depth=5, gamma=0, colsample_bytree=1, subsample=1, nrounds=1000, eta=0.01, min_child_weight=30



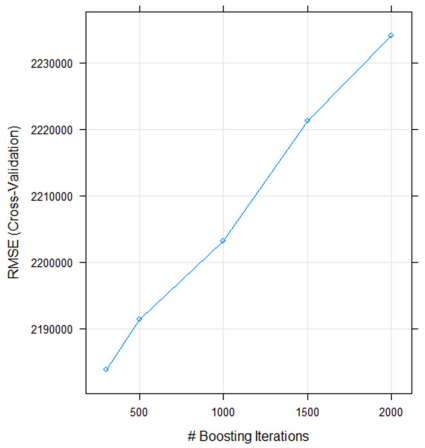
Parada anticipada: Requiere parada anticipada en la iteración 300.



Conjunto aleatoria: Los parámetros que minimizan el error para este conjunto de variables son: max_depth=5, gamma=0, colsample_bytree=1, subsample=1, nrounds=1000, eta=0.01, min_child_weight=30



Parada anticipada: Requiere parada anticipada en la iteración 300.



ANEXOS CÓDIGOS UTILIZADOS EN R Y SAS BASE

Códigos utilizados en R

Modelo de dos partes Variable objetivo binaria (Parte 1)
Redes - Regresión

```
#CARGA LIBRERIAS
library(sas7bdat)
library(nnet)
library(dummies)
library(MASS)
library(reshape)
library(caret)
library(pROC)
library(dplyr)

#DEFINE EL DIRECTORIO
setwd("C:/ ")

# Lectura y esquema de variables
costoBinario<-read.sas7bdat("C:/Costobinario.sas7bdat")
```

```
#costoBinario<-costoBinario[,-35] #Quita la columna ID

#SE CONVIERTE LA VARIABLE OBJETIVO A YES O NO
costoBinario$costo_binario<-
ifelse(costoBinario$costo_binario==1,"Yes","No")

#Resumen base de datos
summary(costoBinario)
#imprime los nombres de las variables
dput(names(costoBinario))
# c("id_afiliado", "costo_binario", "edad", "dias_afiliacion",
#   "dias_afil_porc", "menos1", "dialisis", "oncologia_adultos",
#   "reumatologia_colageno", "VIH", "zona_1", "zona_2", "zona_4",
#   "zona_5", "zona_9", "zona_10", "zona_11", "edad2", "edad_F",
#   "edad_M", "TI_edad21", "TI_edad22", "TI_G_enf_totales1",
"TI_dias_afil1",
#   "TI_dias_afil2", "TI_OPT_edad21", "TI_OPT_edad22",
"TI_OPT_edad24",
#   "TI_enf_totales1", "TI_enf_totales2", "TI_estado_afiliado1",
#   "genero_F", "TI_tipo1", "TI_tipo2", "TI_OPT_edad1",
"TI_OPT_edad2",
#   "TI_OPT_edad3", "TI_OPT_edad4")

#definicion de variables continuas y categoricas
continuas<-c("edad", "dias_afiliacion", "edad2",
"edad_F","edad_M","dias_afil_porc")
categoricas<-c("menos1", "dialisis", "oncologia_adultos",
"reumatologia_colageno", "VIH","zona_1",
"zona_2", "zona_4", "zona_5", "zona_9", "zona_10", "zona_11",
"TI_edad21", "TI_edad22", "TI_G_enf_totales1","TI_dias_afil1",
"TI_dias_afil2", "TI_OPT_edad21", "TI_OPT_edad22",
"TI_OPT_edad24", "TI_enf_totales1", "TI_enf_totales2",
"TI_estado_afiliado1", "genero_F", "TI_tipo1", "TI_tipo2")


# estandarizar las variables continuas

# Calculo medias y desviacion tipica de datos y estandarizo (solo las
continuas)
means <-apply(costoBinario[,continuas],2,mean)
sds<-sapply(costoBinario[,continuas],sd)

# Estandarizo solo las continuas y uno con las categoricas
costoBinariobis<-scale(costoBinario[,continuas], center = means, scale
= sds)
numerocont<-which(colnames(costoBinario)%in%continuas)
costoBinariobis<-cbind(costoBinariobis,costoBinario[,-numerocont])

# El archivo ya esta preparado:no hay missing, las continuas salvo la
dependiente
# estan estandarizadas y las categoricas pasadas a dummy

dput(names(costoBinariobis))
# NOTA: En los modelos se pone solo k-1 dummies por cada categorica

databis<-costoBinariobis
# ********************************
# CRUZADA LOGISTICA
# ********************************
```

```
cruzadalogistica <- function(data=data,vardep=NULL,
listconti=NULL,listclass=NULL,grupos=4,sinicio=1234,repe=5)
{

if  (listclass !=c(""))
{
for (i in 1:dim(array(listclass))) {
numindi<-which(names(data)==listclass[[i]])
data[,numindi]<-as.character(data[,numindi])
data[,numindi]<-as.factor(data[,numindi])
}
}

data[,vardep]<-as.factor(data[,vardep])

# Creo la formula para la logistica

if  (listclass!=c(""))
{
koko<-c(listconti,listclass)
}  else   {
koko<-c(listconti)
}

modelo<-paste(koko,sep="",collapse="+")
formu<-formula(paste(vardep,"~",modelo,sep=""))

formu
# Preparo caret

set.seed(sinicio)
control<-trainControl(method =
"repeatedcv",number=grupos,repeats=repe,
savePredictions = "all",classProbs=TRUE)

# Aplico caret y construyo modelo

regresion <- train(formu,data=data,
trControl=control,method="glm",family = binomial(link="logit"))
preditest<-regresion$pred

preditest$prueba<-strsplit(preditest$Resample,"[.]")
preditest$Fold <- sapply(preditest$prueba, "[", 1)
preditest$Rep <- sapply(preditest$prueba, "[", 2)
preditest$prueba<-NULL

tasafallos<-function(x,y) {
confu<-confusionMatrix(x,y)
tasa<-confu[[3]][1]
return(tasa)
}

# Aplicamos funcion sobre cada Repeticion

medias<-preditest %>%
group_by(Rep) %>%
summarize(tasa=1-tasafallos(pred,obs))

# CalculamoS AUC  por cada Repeticion de cv
# Definimnos funciÃ³n
```

```
auc<-function(x,y) {
curvaroc<-roc(response=x,predictor=y)
auc<-curvaroc$auc
return(auc)
}

# Aplicamos funcion sobre cada Repeticion

mediasbis<-preditest %>%
group_by(Rep) %>%
summarize(auc=auc(obs,Yes))

# Unimos la info de auc y de tasafallos

medias$auc<-mediasbis$auc

return(medias)

}

#Seleccion Miner

medias1<-cruzadalogistica(data=databis,
vardep="costo_binario",listconti=c("edad_F","edad"),
listclass=c("TI_dias_afil1","genero_F", "TI_tipo2",
"TI_estado_afiliado1",
"TI_edad21",  "TI_G_enf_totales1", "TI_OPT_edad2",  "TI_OPT_edad3",
  "zona_9",
"zona_2", "zona_4", "zona_5"),
grupos=4,sinicio=1234,repe=200)

medias1$modelo="reg1"

#Importancia de la variable
medias2<-cruzadalogistica(data=databis,
vardep="costo_binario",listconti=c("edad", "edad2",  "edad_M",
  "edad_F"),
listclass=c("TI_edad21", "TI_G_enf_totales1", "TI_enf_totales1",
"TI_enf_totales2"),
grupos=4,sinicio=1234,repe=200)

medias2$modelo="reg2"

#Seleccion aleatoria 1

medias3<-cruzadalogistica(data=databis,
vardep="costo_binario",listconti=c("dias_afiliacion",
"dias_afil_porc"),
listclass=c("TI_edad21","TI_G_enf_totales1","TI_dias_afil1",
"TI_dias_afil2","TI_estado_afiliado1","genero_F","TI_tipo2",
"zona_1","zona_4","zona_5","zona_9","TI_OPT_edad1","TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=200)

medias3$modelo="reg3"

#Seleccion aleatoria 2
medias4<-cruzadalogistica(data=databis,
vardep="costo_binario",listconti=c("dias_afiliacion","dias_afil_porc")
,
listclass=c("TI_edad21","TI_edad22","TI_G_enf_totales1","TI_estado_afi
liado1",
```

```
"genero_F","TI_tipo2","zona_1","zona_4","zona_5","zona_9","TI_OPT_edad
1","TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=200)

medias4$modelo="reg4"


#mejor con 6
medias5<-cruzadalogistica(data=databis,
vardep="costo_binario",listconti=c("edad_M"),
listclass=c("TI_edad21","TI_G_enf_totales1","TI_dias_afil1","TI_tipo2"
, "zona_9"),
grupos=4,sinicio=1234,repe=200)
medias5$modelo="reg5"

#mejor con 7
medias6<-cruzadalogistica(data=databis,
vardep="costo_binario",listconti=c(""),
listclass=c("TI_G_enf_totales1","TI_dias_afil1","genero_F",
"TI_tipo2","zona_9","TI_OPT_edad2","TI_OPT_edad3"),
grupos=4,sinicio=1234,repe=200)
medias6$modelo="reg6"

#mejor con 8
medias7<-cruzadalogistica(data=databis,
vardep="costo_binario",listconti=c("edad_M","edad_F","edad2"),
listclass=c("TI_G_enf_totales1","TI_dias_afil1","TI_tipo2",
"zona_9","TI_OPT_edad2"),
grupos=4,sinicio=1234,repe=200)
medias7$modelo="reg7"

#mejor con 9
medias8<-cruzadalogistica(data=databis,
vardep="costo_binario",listconti=c("edad_M","edad_F","edad2"),
listclass=c("TI_G_enf_totales1","TI_dias_afil1","TI_tipo2",
"zona_9","TI_OPT_edad2","zona_5"),
grupos=4,sinicio=1234,repe=200)
medias8$modelo="reg8"

#mejor con 10
medias9<-cruzadalogistica(data=databis,
vardep="costo_binario",listconti=c("edad_M","edad_F","edad2"),
listclass=c("TI_G_enf_totales1","TI_dias_afil1","TI_tipo2",
"zona_9","TI_OPT_edad2","zona_5", "zona_1"),
grupos=4,sinicio=1234,repe=200)
medias9$modelo="reg9"

#se calcula el valor promedio de la tasa de fallos y del AUC para cada
conjunto
meansT <-apply(medias8[,2],2,mean)
meansA <-apply(medias4[,3],2,mean)


union1<-rbind(medias1, medias2, medias3, medias4, medias5, medias6,
medias7, medias8, medias9)
par(cex.axis=0.8)
boxplot(data=union1,tasa~modelo,main="TASA FALLOS", font=2,
cex.axis=1.2, col="gray",xlab="", ylab="")
boxplot(data=union1,auc~modelo,main="AUC", cex.axis=1.2,
col="gray",xlab="", ylab="")
```

```
# ***************************
# TUNING CON CARET
# ***************************

# APLICAMOS VALIDACION CRUZADA REPETIDA
# Importante classProbs=TRUE para guardar las probabilidades
# y definir la variable de salida con valores alfanumericos Yes, No

# Validacion cruzada repetida
set.seed(12346)
control<-trainControl(method = "repeatedcv",number=4,repeats=10,
savePredictions = "all",classProbs=TRUE)


# ************************************************************
# avNNet: parametros
# Number of Hidden Units (size, numeric)
# Weight Decay (decay, numeric)
# Bagging (bag, logical)
# ************************************************************


avnnetgrid <-
expand.grid(size=c(3,5,7,8,9,10,11),decay=c(0.01,0.1,0.001),bag=FALSE)

#Tuning con el Set Seleccion Miner
redavnnet<-
train(costo_binario~edad_F+edad+TI_dias_afil1+genero_F+TI_tipo2+TI_est
ado_afiliado1+
TI_edad21+TI_G_enf_totales1+TI_OPT_edad2+TI_OPT_edad3+zona_9+zona_2+zo
na_4+zona_5,
data=costoBinariobis,
method="avNNet",linout =
FALSE,maxit=100,trControl=control,tuneGrid=avnnetgrid,
repeats=10)

redavnnet

#RESULTADO TUNEADO PARA SELECCION MINER: size 3  decay  0.01

#Tuning con Set importancia de la variable
avnnetgrid <-
expand.grid(size=c(3,5,7,9,11,13,15,17,18),decay=c(0.01,0.1,0.001),bag
=FALSE)

redavnnet_2<-
train(costo_binario~edad+edad2+edad_M+edad_F+TI_edad21+TI_G_enf_totale
s1+
TI_enf_totales1+TI_enf_totales2,
data=costoBinariobis,
method="avNNet",linout =
FALSE,maxit=100,trControl=control,tuneGrid=avnnetgrid,
repeats=10)

redavnnet_2

#RESULTADO TUNEADO PARA IMPORTANCIA: size  3 decay 0.001
```

```
#Tuning con set random select 1
avnnetgrid <-
expand.grid(size=c(3,5,7,8,9,10,11),decay=c(0.01,0.1,0.001),bag=FALSE)

redavnnet_3<-
train(costo_binario~dias_afiliacion+TI_edad21+TI_G_enf_totales1+TI_dia
s_afil1+
TI_dias_afil2+TI_estado_afiliado1+genero_F+TI_tipo2+zona_1+zona_4+zona
_5+
zona_9+TI_OPT_edad1+ TI_OPT_edad4,
data=costoBinariobis,
method="avNNet",linout =
FALSE,maxit=100,trControl=control,tuneGrid=avnnetgrid,
repeats=10)

redavnnet_3

#RESULTADO TUNEADO PARA RANDOM1: size 3 decay 0.1

#Tuning con set random select 2
avnnetgrid <-
expand.grid(size=c(3,5,7,8,9,10,11,12),decay=c(0.01,0.1,0.001),bag=FAL
SE)

redavnnet_4<-
train(costo_binario~dias_afil_porc+TI_edad21+TI_edad22+TI_G_enf_totale
s1+
TI_estado_afiliado1+genero_F+TI_tipo2+zona_1+zona_4+zona_5+
zona_9+TI_OPT_edad1+TI_OPT_edad4,
data=costoBinariobis,
method="avNNet",linout =
FALSE,maxit=100,trControl=control,tuneGrid=avnnetgrid,
repeats=10)

redavnnet_4

#RESULTADO TUNEADO PARA RANDOM2: size 3  decay 0.1

# mejor con 10 variables
avnnetgrid <-
expand.grid(size=c(3,5,7,8,9,10,12,14,15),decay=c(0.01,0.1,0.001),bag=
FALSE)

redavnnet_5<-
train(costo_binario~edad_F+edad_M+edad2+TI_G_enf_totales1+TI_dias_afil
1+
TI_tipo2+zona_1+zona_5+zona_9+TI_OPT_edad2,
data=costoBinariobis,
method="avNNet",linout =
FALSE,maxit=100,trControl=control,tuneGrid=avnnetgrid,
repeats=10)

redavnnet_5

#RESULTADO TUNEADO PARA MEJOR CON 10: size 3  decay 0.001



# *********************************
# CRUZADA avNNet
# **************
```

```
cruzadaavnnetbin<-
function(data=data,vardep="vardep",
listconti="listconti",listclass="listclass",grupos=4,sinicio=1234,repe
=5,
size=c(5),decay=c(0.01),repeticiones=5,itera=100)
{

# PreparaciÃ³n del archivo

# b)pasar las categÃ³ricas a dummies

if  (listclass!=c(""))
{
databis<-data[,c(vardep,listconti,listclass)]
databis<- dummy.data.frame(databis, listclass, sep = ".")
}  else   {
databis<-data[,c(vardep,listconti)]
}

# c)estandarizar las variables continuas

# Calculo medias y dtipica de datos y estandarizo (solo las continuas)

means <-apply(databis[,listconti],2,mean)
sds<-sapply(databis[,listconti],sd)

# Estandarizo solo las continuas y uno con las categoricas

datacon<-scale(databis[,listconti], center = means, scale = sds)
numerocont<-which(colnames(databis)%in%listconti)
databis<-cbind(datacon,databis[,-numerocont,drop=FALSE ])

databis[,vardep]<-as.factor(databis[,vardep])

formu<-formula(paste(vardep,"~.",sep=""))

# Preparo caret

set.seed(sinicio)
control<-trainControl(method =
"repeatedcv",number=grupos,repeats=repe,
savePredictions = "all",classProbs=TRUE)

# Aplico caret y construyo modelo

avnnetgrid <-  expand.grid(size=size,decay=decay,bag=FALSE)

avnnet<- train(formu,data=databis,
method="avNNet",linout = FALSE,maxit=itera,repeats=repeticiones,
trControl=control,tuneGrid=avnnetgrid)

print(avnnet$results)

preditest<-avnnet$pred

preditest$prueba<-strsplit(preditest$Resample,"[.]")
preditest$Fold <- sapply(preditest$prueba, "[", 1)
preditest$Rep <- sapply(preditest$prueba, "[", 2)
preditest$prueba<-NULL
```

```
tasafallos<-function(x,y) {
confu<-confusionMatrix(x,y)
tasa<-confu[[3]][1]
return(tasa)
}

# Aplicamos funciÃ³n sobre cada RepeticiÃ³n

medias<-preditest %>%
group_by(Rep) %>%
summarize(tasa=1-tasafallos(pred,obs))

# CalculamoS AUC  por cada RepeticiÃ³n de cv
# Definimnos funciÃ³n

auc<-function(x,y) {
curvaroc<-roc(response=x,predictor=y)
auc<-curvaroc$auc
return(auc)
}

# Aplicamos funciÃ³n sobre cada RepeticiÃ³n

mediasbis<-preditest %>%
group_by(Rep) %>%
summarize(auc=auc(obs,Yes))

# Unimos la info de auc y de tasafallos

medias$auc<-mediasbis$auc

return(medias)

}



summary(costoBinario)
dput(names(costoBinario))

listconti<-c("edad", "dias_afiliacion", "dias_afil_porc","edad2",
"edad_F","edad_M")
listclass<-c("menos1", "dialisis", "oncologia_adultos",
"reumatologia_colageno", "VIH", "zona_1", "zona_2", "zona_4",
"zona_5", "zona_9", "zona_10", "zona_11",
"TI_edad21", "TI_edad22", "TI_G_enf_totales1", "TI_dias_afil1",
"TI_dias_afil2", "TI_OPT_edad21", "TI_OPT_edad22", "TI_OPT_edad24",
"TI_enf_totales1", "TI_enf_totales2", "TI_estado_afiliado1",
"genero_F", "TI_tipo1", "TI_tipo2", "TI_OPT_edad1", "TI_OPT_edad2",
"TI_OPT_edad3", "TI_OPT_edad4")

vardep<-c("costo_binario")


# MINER
medias11<-cruzadaavnnetbin(data=databis,
vardep="costo_binario",listconti=c("edad_F", "edad"),
listclass=c("TI_dias_afil1","genero_F","TI_tipo2","TI_estado_afiliado1
","TI_edad21",
```

```
"TI_G_enf_totales1","TI_OPT_edad2","TI_OPT_edad3","zona_9","zona_2","z
ona_4","zona_5"),
grupos=4,sinicio=12347,repe=200,
size=c(3),decay=c(0.01),repeticiones=10,itera=200)

medias11$modelo="red"


#IMPORTANCIA 5
medias12<-cruzadaavnnetbin(data=databis,
vardep="costo_binario",listconti=c("edad", "edad2",  "edad_M",
  "edad_F"),
listclass=c("TI_edad21","TI_G_enf_totales1","TI_enf_totales1","TI_enf_
totales2"),
grupos=4,sinicio=1234,repe=200,
size=c(5),decay=c(0.001),repeticiones=10,itera=200)

medias12$modelo="red2"

#IMPORTANCIA 3
medias13<-cruzadaavnnetbin(data=databis,
vardep="costo_binario",listconti=c("edad", "edad2",  "edad_M",
  "edad_F"),
listclass=c("TI_edad21","TI_G_enf_totales1","TI_enf_totales1","TI_enf_
totales2"),
grupos=4,sinicio=1234,repe=200,
size=c(3),decay=c(0.001),repeticiones=10,itera=200)

medias13$modelo="red3"

#RANDOM SELECT 1
medias14<-cruzadaavnnetbin(data=costoBinario,
vardep="costo_binario",listconti=c("dias_afiliacion","dias_afil_porc")
,
listclass=c("TI_edad21","TI_G_enf_totales1","TI_dias_afil1","TI_dias_a
fil2","TI_estado_afiliado1","genero_F","TI_tipo2","zona_1","zona_4","z
ona_5","zona_9","TI_OPT_edad1","TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=200,
size=c(3),decay=c(0.1),repeticiones=10,itera=200)

medias14$modelo="red4"

#RANDOM SELECT 2 con 3
medias15<-cruzadaavnnetbin(data=databis,
vardep="costo_binario",listconti=c("dias_afiliacion","dias_afil_porc")
,
listclass=c("TI_edad21","TI_edad22","TI_G_enf_totales1","TI_estado_afi
liado1","genero_F",
"TI_tipo2", "zona_1", "zona_4", "zona_5", "zona_9", "TI_OPT_edad1",
  "TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=200,
size=c(3),decay=c(0.01),repeticiones=10,itera=200)

medias15$modelo="red5"

#aleatorio 2 con 7
medias16<-cruzadaavnnetbin(data=databis,
vardep="costo_binario",listconti=c("dias_afiliacion","dias_afil_porc")
,
listclass=c("TI_edad21","TI_edad22","TI_G_enf_totales1","TI_estado_afi
liado1","genero_F",
```

```
"TI_tipo2", "zona_1", "zona_4", "zona_5", "zona_9", "TI_OPT_edad1",
  "TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=200,
size=c(7),decay=c(0.01),repeticiones=10,itera=200)

medias16$modelo="red6"

#MEJOR CON 10
medias17<-cruzadaavnnetbin(data=databis,
vardep="costo_binario",listconti=c("edad_F", "edad_M", "edad2"),
listclass=c("TI_G_enf_totales1","TI_dias_afil1","TI_tipo2","zona_1",
"zona_5", "zona_9", "TI_OPT_edad2"),
grupos=4,sinicio=1234,repe=200,
size=c(3),decay=c(0.001),repeticiones=10,itera=200)

medias17$modelo="red7"

union1<-rbind(medias11, medias12, medias13, medias14, medias15,
medias16, medias17)

par(cex.axis=0.8)
boxplot(data=union1,tasa~modelo,main="TASA FALLOS", font=2,
cex.axis=1.2, col="gray",xlab="", ylab="")
boxplot(data=union1,auc~modelo,main="AUC", cex.axis=1,
col="gray",xlab="", ylab="")
```

Bagging – Random forest

```
setwd("C:/")
source ("cruzada rf binaria.R")
library(dplyr)
library(pROC)
library(randomForest)
library(caret)
library(dummies)

# SE USA DIRECTAMENTE EL PAQUETE randomForest

set.seed(12345)

#Seleccion Miner definicion numero de arboles
rfbis<-randomForest(factor(costo_binario)~edad_F+ edad+  TI_dias_afil1+
  genero_F+ TI_tipo2+ TI_estado_afiliado1+  TI_edad21+
  TI_G_enf_totales1+ TI_OPT_edad2+  TI_OPT_edad3+  zona_9+zona_2+
  zona_4+zona_5,
data=costoBinariobis,
mtry=14,ntree=10000,sampsize=3000,nodesize=30,replace=TRUE)

plot(rfbis$err.rate[,1], main="Tasa de fallos")


#Seleccion importancia definicion numero de arboles
rfbis2<-
randomForest(factor(costo_binario)~edad+edad2+edad_M+edad_F+TI_edad21+
TI_G_enf_totales1+TI_enf_totales1+TI_enf_totales2,
data=costoBinariobis,
mtry=8,ntree=10000,sampsize=3000,nodesize=30,replace=TRUE)

plot(rfbis2$err.rate[,1], main="Tasa de fallos")


#Seleccion aleatoria 1 definicion numero de arboles
```

```
rfbis3<-randomForest(factor(costo_binario)~dias_afiliacion+
  dias_afil_porc+  TI_edad21+  TI_G_enf_totales1+ TI_dias_afil1+
  TI_dias_afil2+TI_estado_afiliado1+  genero_F+ TI_tipo2+ zona_1+
  zona_4+zona_5+zona_9+TI_OPT_edad1+  TI_OPT_edad4,
data=costoBinariobis,
mtry=15,ntree=10000,sampsize=3000,nodesize=30,replace=TRUE)

plot(rfbis3$err.rate[,1], main="Tasa de fallos")


#Seleccion aleatoria 2 definicion numero de arboles
rfbis4<-randomForest(factor(costo_binario)~dias_afil_porc+
  dias_afiliacion+ TI_edad21+  TI_edad22+  TI_G_enf_totales1+
  TI_estado_afiliado1+  genero_F+ TI_tipo2+ zona_1+zona_4+zona_5+
  zona_9+TI_OPT_edad1+  TI_OPT_edad4,
data=costoBinariobis,
mtry=14,ntree=10000,sampsize=3000,nodesize=30,replace=TRUE)

plot(rfbis4$err.rate[,1], main="Tasa de fallos")


#Seleccion mejor con 10 definicion numero de arboles
rfbis5<-randomForest(factor(costo_binario)~edad_F+  edad_M+edad2+
  TI_G_enf_totales1+ TI_dias_afil1+TI_tipo2+ zona_1+zona_5+zona_9+
  TI_OPT_edad2,
data=costoBinariobis,
mtry=10,ntree=10000,sampsize=3000,nodesize=30,replace=TRUE)

plot(rfbis5$err.rate[,1], main="Tasa de fallos")

# La funcion cruzadarfbin permite plantear bagging
# (para bagging hay que poner mtry=numero de variables independientes)


# Probamos variaciones sobre el tamano muestral en bagging
# con 10 grupos de CV, maximo 563 sampsize


control<-trainControl(method = "cv",number=4,savePredictions = "all",
classProbs=TRUE)

#Seleccion Miner
medias20<-cruzadarfbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("edad_F","edad"),
listclass=c("TI_dias_afil1","genero_F", "TI_tipo2",
"TI_estado_afiliado1",
"TI_edad21",  "TI_G_enf_totales1", "TI_OPT_edad2",  "TI_OPT_edad3",
  "zona_9",
"zona_2", "zona_4", "zona_5"),
grupos=4,sinicio=1234,repe=20,nodesize=30,
mtry=14,ntree=5500,replace=TRUE, sampsize=2000)

medias20$modelo="bag"


medias21<-cruzadarfbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("edad_F","edad"),
listclass=c("TI_dias_afil1","genero_F", "TI_tipo2",
"TI_estado_afiliado1",
"TI_edad21",  "TI_G_enf_totales1", "TI_OPT_edad2",  "TI_OPT_edad3",
  "zona_9",
```

```
"zona_2", "zona_4", "zona_5"),
grupos=4,sinicio=1234,repe=20,nodesize=30,
mtry=14,ntree=5500,replace=TRUE, sampsize=3000)

medias21$modelo="bag2"


medias22<-cruzadarfbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("edad_F","edad"),
listclass=c("TI_dias_afil1","genero_F", "TI_tipo2",
"TI_estado_afiliado1",
"TI_edad21",  "TI_G_enf_totales1", "TI_OPT_edad2",  "TI_OPT_edad3",
  "zona_9",
"zona_2", "zona_4", "zona_5"),
grupos=4,sinicio=1234,repe=20,nodesize=30,
mtry=14,ntree=5500,replace=TRUE, sampsize=4000)

medias22$modelo="bag3"


#importancia


medias23<-cruzadarfbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("edad",  "edad2",  "edad_M", "edad_F"),
listclass=c("TI_edad21", "TI_G_enf_totales1", "TI_enf_totales1",
"TI_enf_totales2"),
grupos=4,sinicio=1234,repe=20,nodesize=30,
mtry=8,ntree=500,replace=TRUE, sampsize=2000)

medias23$modelo="bag4"


medias24<-cruzadarfbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("edad",  "edad2",  "edad_M", "edad_F"),
listclass=c("TI_edad21", "TI_G_enf_totales1", "TI_enf_totales1",
"TI_enf_totales2"),
grupos=4,sinicio=1234,repe=20,nodesize=30,
mtry=8,ntree=500,replace=TRUE, sampsize=3000)

medias24$modelo="bag5"


medias25<-cruzadarfbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("edad",  "edad2",  "edad_M", "edad_F"),
listclass=c("TI_edad21", "TI_G_enf_totales1", "TI_enf_totales1",
"TI_enf_totales2"),
grupos=4,sinicio=1234,repe=20,nodesize=30,
mtry=8,ntree=500,replace=TRUE, sampsize=4000)

medias25$modelo="bag6"

#aleatoria 1

medias26<-cruzadarfbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("dias_afiliacion", "dias_afil_porc"),
listclass=c("TI_edad21","TI_G_enf_totales1","TI_dias_afil1",
"TI_dias_afil2","TI_estado_afiliado1","genero_F","TI_tipo2",
"zona_1","zona_4","zona_5","zona_9","TI_OPT_edad1","TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=20,nodesize=30,
mtry=15,ntree=500,replace=TRUE, sampsize=2000)
```

```
medias26$modelo="bag7"


medias27<-cruzadarfbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("dias_afiliacion", "dias_afil_porc"),
listclass=c("TI_edad21","TI_G_enf_totales1","TI_dias_afil1",
"TI_dias_afil2","TI_estado_afiliado1","genero_F","TI_tipo2",
"zona_1","zona_4","zona_5","zona_9","TI_OPT_edad1","TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=20,nodesize=30,
mtry=15,ntree=500,replace=TRUE, sampsize=3000)

medias27$modelo="bag8"

medias28<-cruzadarfbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("dias_afiliacion", "dias_afil_porc"),
listclass=c("TI_edad21","TI_G_enf_totales1","TI_dias_afil1",
"TI_dias_afil2","TI_estado_afiliado1","genero_F","TI_tipo2",
"zona_1","zona_4","zona_5","zona_9","TI_OPT_edad1","TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=20,nodesize=30,
mtry=15,ntree=500,replace=TRUE, sampsize=4000)

medias28$modelo="bag9"

#Aleatoria 2

medias29<-cruzadarfbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("dias_afiliacion", "dias_afil_porc"),
listclass=c("TI_edad21","TI_edad22","TI_G_enf_totales1","TI_estado_afi
liado1",
"genero_F","TI_tipo2","zona_1","zona_4","zona_5","zona_9","TI_OPT_edad
1","TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=20,nodesize=30,
mtry=14,ntree=1000,replace=TRUE, sampsize=2000)

medias29$modelo="bag10"

medias30<-cruzadarfbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("dias_afiliacion", "dias_afil_porc"),
listclass=c("TI_edad21","TI_edad22","TI_G_enf_totales1","TI_estado_afi
liado1",
"genero_F","TI_tipo2","zona_1","zona_4","zona_5","zona_9","TI_OPT_edad
1","TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=20,nodesize=30,
mtry=14,ntree=1000,replace=TRUE, sampsize=3000)

medias30$modelo="bag11"

medias31<-cruzadarfbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("dias_afiliacion", "dias_afil_porc"),
listclass=c("TI_edad21","TI_edad22","TI_G_enf_totales1","TI_estado_afi
liado1",
"genero_F","TI_tipo2","zona_1","zona_4","zona_5","zona_9","TI_OPT_edad
1","TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=20,nodesize=30,
mtry=14,ntree=1000,replace=TRUE, sampsize=4000)

medias31$modelo="bag12"


#mejor con 10
```

```
medias32<-cruzadarfbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("edad_M","edad_F","edad2"),
listclass=c("TI_G_enf_totales1","TI_dias_afil1","TI_tipo2",
"zona 9","TI OPT edad2","zona 5", "zona 1"),
grupos=4,sinicio=1234,repe=20,nodesize=30,
mtry=10,ntree=6000,replace=TRUE, sampsize=2000)

medias32$modelo="bag13"


medias33<-cruzadarfbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("edad_M","edad_F","edad2"),
listclass=c("TI_G_enf_totales1","TI_dias_afil1","TI_tipo2",
"zona_9","TI_OPT_edad2","zona_5", "zona_1"),
grupos=4,sinicio=1234,repe=20,nodesize=30,
mtry=10,ntree=6000,replace=TRUE, sampsize=3000)

medias33$modelo="bag14"


medias34<-cruzadarfbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("edad_M","edad_F","edad2"),
listclass=c("TI_G_enf_totales1","TI_dias_afil1","TI_tipo2",
"zona_9","TI_OPT_edad2","zona_5", "zona_1"),
grupos=4,sinicio=1234,repe=20,nodesize=30,
mtry=10,ntree=6000,replace=TRUE, sampsize=4000)

medias34$modelo="bag15"

#MINER CON 20 OBS
medias35<-cruzadarfbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("edad_F","edad"),
listclass=c("TI_dias_afil1","genero_F", "TI_tipo2",
"TI_estado_afiliado1",
"TI_edad21",  "TI_G_enf_totales1", "TI_OPT_edad2",  "TI_OPT_edad3",
  "zona_9",
"zona_2", "zona_4", "zona_5"),
grupos=4,sinicio=1234,repe=20,nodesize=20,
mtry=14,ntree=5500,replace=TRUE, sampsize=2000)

medias35$modelo="bag16"


#IMPORTANCIA CON 20
medias36<-cruzadarfbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("edad",  "edad2",  "edad_M", "edad_F"),
listclass=c("TI_edad21", "TI_G_enf_totales1", "TI_enf_totales1",
"TI_enf_totales2"),
grupos=4,sinicio=1234,repe=20,nodesize=20,
mtry=8,ntree=500,replace=TRUE, sampsize=3000)

medias36$modelo="bag17"


#ALEATORIA 1 CON 20
medias37<-cruzadarfbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("dias_afiliacion", "dias_afil_porc"),
listclass=c("TI_edad21","TI_G_enf_totales1","TI_dias_afil1",
"TI_dias_afil2","TI_estado_afiliado1","genero_F","TI_tipo2",
"zona_1","zona_4","zona_5","zona_9","TI_OPT_edad1","TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=20,nodesize=20,
```

```
mtry=15,ntree=500,replace=TRUE, sampsize=2000)

medias37$modelo="bag18"

#ALEATORIA 2 CON 20
medias38<-cruzadarfbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("dias_afiliacion", "dias_afil_porc"),
listclass=c("TI_edad21","TI_edad22","TI_G_enf_totales1","TI_estado_afi
liado1",
"genero_F","TI_tipo2","zona_1","zona_4","zona_5","zona_9","TI_OPT_edad
1","TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=20,nodesize=20,
mtry=14,ntree=1000,replace=TRUE, sampsize=2000)

medias38$modelo="bag19"

#MEJOR CON 10

medias39<-cruzadarfbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("edad_M","edad_F","edad2"),
listclass=c("TI_G_enf_totales1","TI_dias_afil1","TI_tipo2",
"zona_9","TI_OPT_edad2","zona_5", "zona_1"),
grupos=4,sinicio=1234,repe=20,nodesize=20,
mtry=10,ntree=6000,replace=TRUE, sampsize=2000)

medias39$modelo="bag20"

meansT <-apply(medias20[,2],2,mean)

union1<-rbind(medias20, medias21, medias22, medias23, medias24,
medias25, medias26, medias27, medias28, medias29, medias30, medias31,
medias32, medias33, medias34, medias35, medias36, medias37, medias38,
medias39)
union1<-rbind(medias35, medias36, medias37, medias38, medias39)

union1<-rbind(medias20, medias24, medias26, medias29, medias32,
medias35, medias38)

#graficos
uni<-union1
uni$modelo <- with(uni,
reorder(modelo,tasa, mean))
par(cex.axis=1.2,las=2)
boxplot(data=uni,tasa~modelo,col="gray", main="Tasa de fallos",
xlab="", ylab="")
boxplot(data=union1,auc~modelo,main="AUC", col="gray", xlab="",
ylab="")


#Random Forest

#MINER CON 12
medias40<-cruzadarfbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("edad F","edad"),
listclass=c("TI_dias_afil1","genero_F", "TI_tipo2",
"TI_estado_afiliado1",
"TI_edad21",  "TI_G_enf_totales1", "TI_OPT_edad2",  "TI_OPT_edad3",
  "zona_9",
"zona_2", "zona_4", "zona_5"),
grupos=4,sinicio=1234,repe=200,nodesize=30,
mtry=12,ntree=5500,replace=TRUE, sampsize=2000)
```

```
medias40$modelo="RF1"

medias41<-cruzadarfbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("edad_F","edad"),
listclass=c("TI_dias_afil1","genero_F", "TI_tipo2",
"TI_estado_afiliado1",
"TI_edad21",  "TI_G_enf_totales1", "TI_OPT_edad2",  "TI_OPT_edad3",
  "zona_9",
"zona_2", "zona_4", "zona_5"),
grupos=4,sinicio=1234,repe=200,nodesize=30,
mtry=10,ntree=5500,replace=TRUE, sampsize=2000)

medias41$modelo="RF2"

medias42<-cruzadarfbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("edad_F","edad"),
listclass=c("TI_dias_afil1","genero_F", "TI_tipo2",
"TI_estado_afiliado1",
"TI_edad21",  "TI_G_enf_totales1", "TI_OPT_edad2",  "TI_OPT_edad3",
  "zona_9",
"zona_2", "zona_4", "zona_5"),
grupos=4,sinicio=1234,repe=200,nodesize=30,
mtry=8,ntree=5500,replace=TRUE, sampsize=2000)

medias42$modelo="RF3"

medias43<-cruzadarfbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("edad_F","edad"),
listclass=c("TI_dias_afil1","genero_F", "TI_tipo2",
"TI_estado_afiliado1",
"TI_edad21",  "TI_G_enf_totales1", "TI_OPT_edad2",  "TI_OPT_edad3",
  "zona_9",
"zona_2", "zona_4", "zona_5"),
grupos=4,sinicio=1234,repe=200,nodesize=30,
mtry=6,ntree=5500,replace=TRUE, sampsize=2000)

medias43$modelo="RF4"


medias44<-cruzadarfbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("edad_F","edad"),
listclass=c("TI_dias_afil1","genero_F", "TI_tipo2",
"TI_estado_afiliado1",
"TI_edad21",  "TI_G_enf_totales1", "TI_OPT_edad2",  "TI_OPT_edad3",
  "zona_9",
"zona_2", "zona_4", "zona_5"),
grupos=4,sinicio=1234,repe=200,nodesize=30,
mtry=4,ntree=5500,replace=TRUE, sampsize=2000)

medias44$modelo="RF5"

#IMPORTANCIA CON 20
medias45<-cruzadarfbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("edad",  "edad2",  "edad_M", "edad_F"),
listclass=c("TI_edad21", "TI_G_enf_totales1", "TI_enf_totales1",
"TI_enf_totales2"),
grupos=4,sinicio=1234,repe=200,nodesize=30,
mtry=7,ntree=500,replace=TRUE, sampsize=3000)

medias45$modelo="RF6"
```

```
medias46<-cruzadarfbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("edad",  "edad2",  "edad M", "edad F"),
listclass=c("TI_edad21", "TI_G_enf_totales1", "TI_enf_totales1",
"TI_enf_totales2"),
grupos=4,sinicio=1234,repe=200,nodesize=30,
mtry=6,ntree=500,replace=TRUE, sampsize=3000)

medias46$modelo="RF7"


medias47<-cruzadarfbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("edad",  "edad2",  "edad_M", "edad_F"),
listclass=c("TI_edad21", "TI_G_enf_totales1", "TI_enf_totales1",
"TI_enf_totales2"),
grupos=4,sinicio=1234,repe=200,nodesize=30,
mtry=5,ntree=500,replace=TRUE, sampsize=3000)

medias47$modelo="RF8"


medias48<-cruzadarfbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("edad",  "edad2",  "edad_M", "edad_F"),
listclass=c("TI_edad21", "TI_G_enf_totales1", "TI_enf_totales1",
"TI_enf_totales2"),
grupos=4,sinicio=1234,repe=200,nodesize=30,
mtry=4,ntree=500,replace=TRUE, sampsize=3000)

medias48$modelo="RF9"

#ALEATORIA 1
medias50<-cruzadarfbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("dias_afiliacion", "dias_afil_porc"),
listclass=c("TI_edad21","TI_G_enf_totales1","TI_dias_afil1",
"TI_dias_afil2","TI_estado_afiliado1","genero_F","TI_tipo2",
"zona_1","zona_4","zona_5","zona_9","TI_OPT_edad1","TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=200,nodesize=30,
mtry=13,ntree=500,replace=TRUE, sampsize=2000)

medias50$modelo="RF10"

medias51<-cruzadarfbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("dias_afiliacion", "dias_afil_porc"),
listclass=c("TI_edad21","TI_G_enf_totales1","TI_dias_afil1",
"TI_dias_afil2","TI_estado_afiliado1","genero_F","TI_tipo2",
"zona_1","zona_4","zona_5","zona_9","TI_OPT_edad1","TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=200,nodesize=30,
mtry=11,ntree=500,replace=TRUE, sampsize=2000)

medias51$modelo="RF11"

medias52<-cruzadarfbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("dias afiliacion", "dias afil porc"),
listclass=c("TI_edad21","TI_G_enf_totales1","TI_dias_afil1",
"TI_dias_afil2","TI_estado_afiliado1","genero_F","TI_tipo2",
"zona_1","zona_4","zona_5","zona_9","TI_OPT_edad1","TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=200,nodesize=30,
mtry=9,ntree=500,replace=TRUE, sampsize=2000)

medias52$modelo="RF12"
```

```
medias53<-cruzadarfbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("dias_afiliacion", "dias_afil_porc"),
listclass=c("TI_edad21","TI_G_enf_totales1","TI_dias_afil1",
"TI_dias_afil2","TI_estado_afiliado1","genero_F","TI_tipo2",
"zona_1","zona_4","zona_5","zona_9","TI_OPT_edad1","TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=200,nodesize=30,
mtry=7,ntree=500,replace=TRUE, sampsize=2000)

medias53$modelo="RF13"



medias54<-cruzadarfbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("dias_afiliacion", "dias_afil_porc"),
listclass=c("TI_edad21","TI_G_enf_totales1","TI_dias_afil1",
"TI_dias_afil2","TI_estado_afiliado1","genero_F","TI_tipo2",
"zona_1","zona_4","zona_5","zona_9","TI_OPT_edad1","TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=200,nodesize=30,
mtry=5,ntree=500,replace=TRUE, sampsize=2000)

medias54$modelo="RF14"

#ALEATORIA 2 CON 20
medias55<-cruzadarfbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("dias_afiliacion", "dias_afil_porc"),
listclass=c("TI_edad21","TI_edad22","TI_G_enf_totales1","TI_estado_afi
liado1",
"genero_F","TI_tipo2","zona_1","zona_4","zona_5","zona_9","TI_OPT_edad
1","TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=200,nodesize=30,
mtry=12,ntree=1000,replace=TRUE, sampsize=2000)

medias55$modelo="RF15"


medias56<-cruzadarfbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("dias_afiliacion", "dias_afil_porc"),
listclass=c("TI_edad21","TI_edad22","TI_G_enf_totales1","TI_estado_afi
liado1",
"genero_F","TI_tipo2","zona_1","zona_4","zona_5","zona_9","TI_OPT_edad
1","TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=200,nodesize=30,
mtry=10,ntree=1000,replace=TRUE, sampsize=2000)

medias56$modelo="RF16"

medias57<-cruzadarfbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("dias_afiliacion", "dias_afil_porc"),
listclass=c("TI_edad21","TI_edad22","TI_G_enf_totales1","TI_estado_afi
liado1",
"genero_F","TI_tipo2","zona_1","zona_4","zona_5","zona_9","TI_OPT_edad
1","TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=200,nodesize=30,
mtry=8,ntree=1000,replace=TRUE, sampsize=2000)

medias57$modelo="RF17"

medias58<-cruzadarfbin(data=costoBinariobis, vardep="costo_binario",
```

```
listconti=c("dias_afiliacion", "dias_afil_porc"),
listclass=c("TI_edad21","TI_edad22","TI_G_enf_totales1","TI_estado_afi
liado1",
"genero_F","TI_tipo2","zona_1","zona_4","zona_5","zona_9","TI_OPT_edad
1","TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=200,nodesize=30,
mtry=6,ntree=1000,replace=TRUE, sampsize=2000)

medias58$modelo="RF18"


medias59<-cruzadarfbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("dias_afiliacion", "dias_afil_porc"),
listclass=c("TI_edad21","TI_edad22","TI_G_enf_totales1","TI_estado_afi
liado1",
"genero_F","TI_tipo2","zona_1","zona_4","zona_5","zona_9","TI_OPT_edad
1","TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=200,nodesize=30,
mtry=4,ntree=1000,replace=TRUE, sampsize=2000)

medias59$modelo="RF19"



#mejor con 10
medias60<-cruzadarfbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("edad_M","edad_F","edad2"),
listclass=c("TI_G_enf_totales1","TI_dias_afil1","TI_tipo2",
"zona_9","TI_OPT_edad2","zona_5", "zona_1"),
grupos=4,sinicio=1234,repe=200,nodesize=30,
mtry=8,ntree=6000,replace=TRUE, sampsize=2000)

medias60$modelo="RF20"

medias61<-cruzadarfbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("edad_M","edad_F","edad2"),
listclass=c("TI_G_enf_totales1","TI_dias_afil1","TI_tipo2",
"zona_9","TI_OPT_edad2","zona_5", "zona_1"),
grupos=4,sinicio=1234,repe=200,nodesize=30,
mtry=6,ntree=6000,replace=TRUE, sampsize=2000)

medias61$modelo="RF21"

medias62<-cruzadarfbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("edad_M","edad_F","edad2"),
listclass=c("TI_G_enf_totales1","TI_dias_afil1","TI_tipo2",
"zona_9","TI_OPT_edad2","zona_5", "zona_1"),
grupos=4,sinicio=1234,repe=200,nodesize=30,
mtry=4,ntree=6000,replace=TRUE, sampsize=2000)

medias62$modelo="RF22"



#Resultados random forest

union1<-rbind(medias40, medias41, medias42, medias43,
medias44,medias45, medias46, medias47, medias48, medias50, medias51,
medias52, medias53, medias54, medias55, medias56, medias57, medias58,
medias59, medias60, medias61, medias62)
union1<-rbind(medias45, medias46, medias47, medias48)
```

```
union1<-rbind(medias50, medias51, medias52, medias53, medias54)
union1<-rbind(medias55, medias56, medias57, medias58, medias59)
union1<-rbind(medias60, medias61, medias62)


par(cex.axis=0.8)
boxplot(data=union1,tasa~modelo,main="TASA FALLOS",col="gray")



uni<-union1
uni$modelo <- with(uni,
reorder(modelo,tasa, mean))
par(cex.axis=1.2,las=2)
boxplot(data=uni,tasa~modelo,col="gray", main="Tasa de fallos",
xlab="", ylab="")

uni<-union1
uni$modelo <- with(uni,
reorder(modelo,auc,mean))
boxplot(data=union1,auc~modelo,main="AUC", col="gray",xlab="",
ylab="")
```

Incremento gradiente

```
#GRADIENT BOOSTING

# TUNEADO DE GRADIENT BOOSTING CON CARET

# Caret permite tunear estos parÃ¡metros bÃ¡sicos:
#
#  shrinkage (parÃ¡metro v de regularizaciÃ³n, mide la velocidad de
ajuste, a menor v, mÃ¡s lento y necesita mÃ¡s iteraciones, pero es
mÃ¡s fino en el ajuste)
#  n.minobsinnode: tamaÃ±o mÃ¡ximo de nodos finales (el principal
parÃ¡metro que mide la complejidad)
#  n.trees=el nÃºmero de iteraciones (Ã¡rboles)
#  interaction.depth (2 para Ã¡rboles binarios)


library(caret)
library(dummies)
library (dplyr)
library (pROC)

set.seed(12345)

gbmgrid<-expand.grid(shrinkage=c(0.1,0.05,0.01,0.001),
n.minobsinnode=c(10,20,30),
n.trees=c(1000,2000,3000),
interaction.depth=c(2))

control<-trainControl(method = "cv",number=4,savePredictions = "all",
classProbs=TRUE)

#MINER
gbm<- train(factor(costo_binario)~edad_F+  edad+  TI_dias_afil1+
  genero_F+ TI_tipo2+ TI_estado_afiliado1+  TI_edad21+
  TI_G_enf_totales1+ TI_OPT_edad2+  TI_OPT_edad3+  zona_9+zona_2+
  zona_4+zona_5,
```

```
data=costoBinariobis,
method="gbm",trControl=control,tuneGrid=gbmgrid,
distribution="bernoulli", bag.fraction=1,verbose=FALSE)

gbm



#IMPORTANCIA
gbm<-
train(factor(costo_binario)~edad+edad2+edad_M+edad_F+TI_edad21+TI_G_en
f_totales1+TI_enf_totales1+TI_enf_totales2,
data=costoBinariobis,
method="gbm",trControl=control,tuneGrid=gbmgrid,
distribution="bernoulli", bag.fraction=1,verbose=FALSE)

gbm

plot(gbm,cex.axis=1.2)


#ALEATORIA 1
gbm<-
train(factor(costo_binario)~dias_afiliacion+dias_afil_porc+TI_edad21+T
I_G_enf_totales1+TI_dias_afil1+
   TI_dias_afil2+TI_estado_afiliado1+genero_F+TI_tipo2+zona_1+zona_4+zo
na_5+zona_9+TI_OPT_edad1+TI_OPT_edad4,
data=costoBinariobis,
method="gbm",trControl=control,tuneGrid=gbmgrid,
distribution="bernoulli", bag.fraction=1,verbose=FALSE)

gbm

plot(gbm,cex.axis=1.2)

#ALEATORIA 2
gbm<- train(factor(costo_binario)~dias_afil_porc+ dias_afiliacion+
   TI_edad21+  TI_edad22+  TI_G_enf_totales1+ TI_estado_afiliado1+
   genero_F+ TI_tipo2+ zona_1+zona_4+zona_5+zona_9+TI_OPT_edad1+
   TI_OPT_edad4,
data=costoBinariobis,
method="gbm",trControl=control,tuneGrid=gbmgrid,
distribution="bernoulli", bag.fraction=1,verbose=FALSE)

gbm

plot(gbm,cex.axis=1.2)


#MEJOR 10
gbm<- train(factor(costo_binario)~edad_F+  edad_M+edad2+
   TI_G_enf_totales1+ TI_dias_afil1+TI_tipo2+ zona_1+zona_5+zona_9+
   TI_OPT_edad2,
data=costoBinariobis,
method="gbm",trControl=control,tuneGrid=gbmgrid,
distribution="bernoulli", bag.fraction=1,verbose=FALSE)

gbm

plot(gbm,cex.axis=1.2)
```

```
# ESTUDIO DE EARLY STOPPING
# Probamos a fijar algunos parÃ¡metros para ver como evoluciona
# en funciÃ³n de las iteraciones

#MINER
gbmgrid<-expand.grid(shrinkage=c(0.01),
n.minobsinnode=c(20),
n.trees=c(1000,2000,5000,6000),
interaction.depth=c(2))

control<-trainControl(method = "cv",number=4,savePredictions = "all",
classProbs=TRUE)

gbm<- train(factor(costo_binario)~edad_F+  edad+  TI_dias_afil1+
   genero_F+ TI_tipo2+ TI_estado_afiliado1+  TI_edad21+
   TI_G_enf_totales1+ TI_OPT_edad2+  TI_OPT_edad3+  zona_9+zona_2+
   zona_4+zona_5,
data=costoBinariobis,
method="gbm",trControl=control,tuneGrid=gbmgrid,
distribution="bernoulli", bag.fraction=1,verbose=FALSE)

plot(gbm)

#IMPORTANCIA
gbmgrid<-expand.grid(shrinkage=c(0.01),
n.minobsinnode=c(10),
n.trees=c(1000,2000,5000,6000),
interaction.depth=c(2))

control<-trainControl(method = "cv",number=4,savePredictions = "all",
classProbs=TRUE)

gbm<-
train(factor(costo_binario)~edad+edad2+edad_M+edad_F+TI_edad21+TI_G_en
f_totales1+TI_enf_totales1+TI_enf_totales2,
data=costoBinariobis,
method="gbm",trControl=control,tuneGrid=gbmgrid,
distribution="bernoulli", bag.fraction=1,verbose=FALSE)

plot(gbm)

#ALEATORIA 1
gbmgrid<-expand.grid(shrinkage=c(0.01),
n.minobsinnode=c(20),
n.trees=c(1000,2000,5000,6000),
interaction.depth=c(2))

control<-trainControl(method = "cv",number=4,savePredictions = "all",
classProbs=TRUE)

gbm<- train(factor(costo_binario)~dias_afiliacion+  dias_afil_porc+
   TI_edad21+  TI_G_enf_totales1+ TI_dias_afil1+TI_dias_afil2+
   TI_estado_afiliado1+  genero_F+ TI_tipo2+ zona_1+zona_4+zona_5+
   zona_9+TI_OPT_edad1+  TI_OPT_edad4,
data=costoBinariobis,
method="gbm",trControl=control,tuneGrid=gbmgrid,
distribution="bernoulli", bag.fraction=1,verbose=FALSE)

plot(gbm)

#ALEATORIA 2
```

```
gbmgrid<-expand.grid(shrinkage=c(0.01),
n.minobsinnode=c(30),
n.trees=c(1000,2000,5000,6000),
interaction.depth=c(2))

control<-trainControl(method = "cv",number=4,savePredictions = "all",
classProbs=TRUE)

gbm<- train(factor(costo_binario)~dias_afil_porc+ dias_afiliacion+
   TI_edad21+  TI_edad22+  TI_G_enf_totales1+ TI_estado_afiliado1+
   genero_F+ TI_tipo2+ zona_1+zona_4+zona_5+zona_9+TI_OPT_edad1+
   TI_OPT_edad4,
data=costoBinariobis,
method="gbm",trControl=control,tuneGrid=gbmgrid,
distribution="bernoulli", bag.fraction=1,verbose=FALSE)

plot(gbm)

#MEJOR CON 10
gbmgrid<-expand.grid(shrinkage=c(0.01),
n.minobsinnode=c(10),
n.trees=c(1000,2000,5000,6000),
interaction.depth=c(2))

control<-trainControl(method = "cv",number=4,savePredictions = "all",
classProbs=TRUE)

gbm<- train(factor(costo_binario)~edad_F+  edad_M+edad2+
   TI_G_enf_totales1+ TI_dias_afil1+TI_tipo2+ zona_1+zona_5+zona_9+
   TI_OPT_edad2,
data=costoBinariobis,
method="gbm",trControl=control,tuneGrid=gbmgrid,
distribution="bernoulli", bag.fraction=1,verbose=FALSE)

plot(gbm)


# IMPORTANCIA DE VARIABLES
# par(cex=1.3)
# summary(gbm)
#
# tabla<-summary(gbm)
# par(cex=0.5,las=2)
# barplot(tabla$rel.inf,names.arg=row.names(tabla))
#
# databis<-costoBinariobis
# data<-costoBinario
# La funciOn cruzadagbmbin permite plantear gradient boosting para
binarias

source ("cruzada gbm binaria.R")
setwd("C:/")
getwd()

#MINER
medias65<-cruzadagbmbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("edad_F","edad"),
listclass=c("TI_dias_afil1","genero_F", "TI_tipo2",
"TI_estado_afiliado1",
"TI_edad21",  "TI_G_enf_totales1", "TI_OPT_edad2",  "TI_OPT_edad3",
   "zona_9",
```

```
"zona_2", "zona_4", "zona_5"),
grupos=4,sinicio=12345,repe=200,
n.minobsinnode=20,shrinkage=0.01,n.trees=2000,interaction.depth=2)

medias65$modelo="gbm"


medias66<-cruzadagbmbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("edad_F","edad"),
listclass=c("TI_dias_afil1","genero_F", "TI_tipo2",
"TI_estado_afiliado1",
"TI_edad21",  "TI_G_enf_totales1", "TI_OPT_edad2",  "TI_OPT_edad3",
  "zona_9",
"zona_2", "zona_4", "zona_5"),
grupos=4,sinicio=12346,repe=200,
n.minobsinnode=20,shrinkage=0.01,n.trees=2000,interaction.depth=2)

medias66$modelo="gbm2"

medias67<-cruzadagbmbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("edad_F","edad"),
listclass=c("TI_dias_afil1","genero_F", "TI_tipo2",
"TI_estado_afiliado1",
"TI_edad21",  "TI_G_enf_totales1", "TI_OPT_edad2",  "TI_OPT_edad3",
  "zona_9",
"zona_2", "zona_4", "zona_5"),
grupos=4,sinicio=12345,repe=200,
n.minobsinnode=20,shrinkage=0.0001,n.trees=5000,interaction.depth=2)

medias67$modelo="gbm3"

#IMPORTANCIA
medias68<-cruzadagbmbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("edad",  "edad2",  "edad_M","edad_F"),
listclass=c("TI_edad21", "TI_G_enf_totales1", "TI_enf_totales1",
"TI_enf_totales2"),
grupos=4,sinicio=12345,repe=200,
n.minobsinnode=10,shrinkage=0.01,n.trees=1000,interaction.depth=2)

medias68$modelo="gbm4"


medias69<-cruzadagbmbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("edad",  "edad2",  "edad_M","edad_F"),
listclass=c("TI_edad21", "TI_G_enf_totales1", "TI_enf_totales1",
"TI_enf_totales2"),
grupos=4,sinicio=12346,repe=200,
n.minobsinnode=10,shrinkage=0.01,n.trees=1000,interaction.depth=2)

medias69$modelo="gbm5"


medias70<-cruzadagbmbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("edad",  "edad2",  "edad M","edad F"),
listclass=c("TI_edad21", "TI_G_enf_totales1", "TI_enf_totales1",
"TI_enf_totales2"),
grupos=4,sinicio=12345,repe=200,
n.minobsinnode=10,shrinkage=0.0001,n.trees=5000,interaction.depth=2)

medias70$modelo="gbm6"
```

```
#ALEATORIA 1
medias71<-cruzadagbmbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("dias afiliacion", "dias afil porc"),
listclass=c("TI_edad21","TI_G_enf_totales1","TI_dias_afil1",
"TI_dias_afil2","TI_estado_afiliado1","genero_F","TI_tipo2",
"zona_1","zona_4","zona_5","zona_9","TI_OPT_edad1","TI_OPT_edad4"),
grupos=4,sinicio=12345,repe=200,
n.minobsinnode=20,shrinkage=0.01,n.trees=2000,interaction.depth=2)

medias71$modelo="gbm7"


medias72<-cruzadagbmbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("dias_afiliacion", "dias_afil_porc"),
listclass=c("TI_edad21","TI_G_enf_totales1","TI_dias_afil1",
"TI_dias_afil2","TI_estado_afiliado1","genero_F","TI_tipo2",
"zona_1","zona_4","zona_5","zona_9","TI_OPT_edad1","TI_OPT_edad4"),
grupos=4,sinicio=12346,repe=200,
n.minobsinnode=20,shrinkage=0.01,n.trees=2000,interaction.depth=2)

medias72$modelo="gbm8"

medias73<-cruzadagbmbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("dias_afiliacion", "dias_afil_porc"),
listclass=c("TI_edad21","TI_G_enf_totales1","TI_dias_afil1",
"TI_dias_afil2","TI_estado_afiliado1","genero_F","TI_tipo2",
"zona_1","zona_4","zona_5","zona_9","TI_OPT_edad1","TI_OPT_edad4"),
grupos=4,sinicio=12345,repe=200,
n.minobsinnode=20,shrinkage=0.0001,n.trees=5000,interaction.depth=2)

medias73$modelo="gbm9"


#ALEATORIA 2
medias74<-cruzadagbmbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("dias_afiliacion", "dias_afil_porc"),
listclass=c("TI_edad21","TI_edad22","TI_G_enf_totales1","TI_estado_afi
liado1",
"genero_F","TI_tipo2","zona_1","zona_4","zona_5","zona_9","TI_OPT_edad
1","TI_OPT_edad4"),
grupos=4,sinicio=12345,repe=200,
n.minobsinnode=30,shrinkage=0.01,n.trees=2000,interaction.depth=2)

medias74$modelo="gbm10"

medias75<-cruzadagbmbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("dias_afiliacion", "dias_afil_porc"),
listclass=c("TI_edad21","TI_edad22","TI_G_enf_totales1","TI_estado_afi
liado1",
"genero_F","TI_tipo2","zona_1","zona_4","zona_5","zona_9","TI_OPT_edad
1","TI_OPT_edad4"),
grupos=4,sinicio=12346,repe=200,
n.minobsinnode=30,shrinkage=0.01,n.trees=2000,interaction.depth=2)

medias75$modelo="gbm11"

medias76<-cruzadagbmbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("dias_afiliacion", "dias_afil_porc"),
listclass=c("TI_edad21","TI_edad22","TI_G_enf_totales1","TI_estado_afi
liado1",
```

```
"genero_F","TI_tipo2","zona_1","zona_4","zona_5","zona_9","TI_OPT_edad
1","TI_OPT_edad4"),
grupos=4,sinicio=12345,repe=200,
n.minobsinnode=30,shrinkage=0.0001,n.trees=5000,interaction.depth=2)

medias76$modelo="gbm12"


#MEJOR CON 10
medias77<-cruzadagbmbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("edad_M","edad_F","edad2"),
listclass=c("TI_G_enf_totales1","TI_dias_afil1","TI_tipo2",
"zona_9","TI_OPT_edad2","zona_5", "zona_1"),
grupos=4,sinicio=12345,repe=200,
n.minobsinnode=10,shrinkage=0.01,n.trees=2000,interaction.depth=2)

medias77$modelo="gbm13"

medias78<-cruzadagbmbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("edad_M","edad_F","edad2"),
listclass=c("TI_G_enf_totales1","TI_dias_afil1","TI_tipo2",
"zona_9","TI_OPT_edad2","zona_5", "zona_1"),
grupos=4,sinicio=12346,repe=200,
n.minobsinnode=10,shrinkage=0.01,n.trees=2000,interaction.depth=2)

medias78$modelo="gbm14"

medias79<-cruzadagbmbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("edad_M","edad_F","edad2"),
listclass=c("TI_G_enf_totales1","TI_dias_afil1","TI_tipo2",
"zona_9","TI_OPT_edad2","zona_5", "zona_1"),
grupos=4,sinicio=12345,repe=200,
n.minobsinnode=10,shrinkage=0.0001,n.trees=5000,interaction.depth=2)

medias79$modelo="gbm15"



#ALEATORIA 2
medias80<-cruzadagbmbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("dias_afiliacion", "dias_afil_porc"),
listclass=c("TI_edad21","TI_edad22","TI_G_enf_totales1","TI_estado_afi
liado1",
"genero_F","TI_tipo2","zona_1","zona_4","zona_5","zona_9","TI_OPT_edad
1","TI_OPT_edad4"),
grupos=4,sinicio=12345,repe=200,
n.minobsinnode=30,shrinkage=0.02,n.trees=2000,interaction.depth=2)

medias80$modelo="gbm16"

medias81<-cruzadagbmbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("dias_afiliacion", "dias_afil_porc"),
listclass=c("TI_edad21","TI_edad22","TI_G_enf_totales1","TI_estado_afi
liado1",
"genero_F","TI_tipo2","zona_1","zona_4","zona_5","zona_9","TI_OPT_edad
1","TI_OPT_edad4"),
grupos=4,sinicio=12345,repe=200,
n.minobsinnode=30,shrinkage=0.015,n.trees=2000,interaction.depth=2)

medias81$modelo="gbm17"
```

```
medias82<-cruzadagbmbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("dias_afiliacion", "dias_afil_porc"),
listclass=c("TI_edad21","TI_edad22","TI_G_enf_totales1","TI_estado_afi
liado1",
"genero_F","TI_tipo2","zona_1","zona_4","zona_5","zona_9","TI_OPT_edad
1","TI_OPT_edad4"),
grupos=4,sinicio=12345,repe=200,
n.minobsinnode=30,shrinkage=0.005,n.trees=2000,interaction.depth=2)

medias82$modelo="gbm18"

union1<-rbind(medias65, medias66, medias67, medias68, medias69,
medias70, medias71, medias72, medias73, medias74, medias75, medias76,
medias77, medias78, medias79, medias80, medias81, medias82)


uni<-union1
uni$modelo <- with(uni,
reorder(modelo,tasa, mean))
par(cex.axis=1.2,las=2)
boxplot(data=uni,tasa~modelo,col="gray", main="Tasa de fallos",
xlab="", ylab="")
boxplot(data=union1,auc~modelo,main="AUC", col="gray", xlab="",
ylab="")
```

Xgboost

```
# EJEMPLOS XGBOOST

# TUNEADO DE XGBOOST CON CARET

# Caret permite tunear estos parÃ¡metros bÃ¡sicos:
#
# nrounds (# Boosting Iterations)
# max_depth (Max Tree Depth)
# eta (Shrinkage)
# gamma (Minimum Loss Reduction)
# colsample_bytree (Subsample Ratio of Columns)
# min_child_weight (Minimum Sum of Instance Weight)
# subsample (Subsample Percentage)

library(caret)
library(dplyr)
library(xgboost)
setwd("C:/")
source ("cruzada xgboost binaria.R")
set.seed(12345)

xgbmgrid<-expand.grid(
min_child_weight=c(10,20,30),
eta=c(0.01,0.015,0.025,0.05,0.1),
nrounds=c(1000,2000,5000),
max_depth=5,gamma=0,colsample_bytree=1,subsample=1)



control<-trainControl(method = "cv",number=4,p=0.7,savePredictions =
"all",
classProbs=TRUE)
```

```
#MINER

xgbmgrid<-expand.grid(
min_child_weight=10,
eta=0.01,
nrounds=1000,
max_depth=
c(5,7,9,12),gamma=0,colsample_bytree=c(0.8,1),subsample=c(0.8,1))

xgbm<- train(factor(costo_binario)~edad_F+ edad+  TI_dias_afil1+
   genero_F+ TI_tipo2+ TI_estado_afiliado1+  TI_edad21+
   TI_G_enf_totales1+ TI_OPT_edad2+  TI_OPT_edad3+  zona_9+zona_2+
   zona_4+zona_5,
data=costoBinariobis,
method="xgbTree",trControl=control,
tuneGrid=xgbmgrid,verbose=FALSE)

xgbm

plot(xgbm)


#IMPORTANCIA
xgbmgrid<-expand.grid(
min_child_weight=20,
eta=0.01,
nrounds=2000,
max_depth=
c(5,7,9,12),gamma=0,colsample_bytree=c(0.8,1),subsample=c(0.8,1))

xgbm2<-
train(factor(costo_binario)~edad+edad2+edad_M+edad_F+TI_edad21+TI_G_en
f_totales1+TI_enf_totales1+TI_enf_totales2,
data=costoBinariobis,
method="xgbTree",trControl=control,
tuneGrid=xgbmgrid,verbose=FALSE)

xgbm2

plot(xgbm2)

#ALEATORIA 1
xgbmgrid<-expand.grid(
min_child_weight=30,
eta=0.015,
nrounds=1000,
max_depth=
c(5,7,9,12),gamma=0,colsample_bytree=c(0.8,1),subsample=c(0.8,1))

xgbm3<- train(factor(costo_binario)~dias_afiliacion+dias_afil_porc+
   TI_edad21+  TI_G_enf_totales1+ TI_dias_afil1+TI_dias_afil2+
   TI_estado_afiliado1+  genero_F+ TI_tipo2+ zona_1+zona_4+zona_5+
   zona_9+TI_OPT_edad1+  TI_OPT_edad4,
data=costoBinariobis,
method="xgbTree",trControl=control,
tuneGrid=xgbmgrid,verbose=FALSE)

xgbm3

plot(xgbm3)
```

```
#ALEATORIA 2
xgbmgrid<-expand.grid(
min child weight=20,
eta=0.015,
nrounds=1000,
max_depth=
c(5,7,9,12),gamma=0,colsample_bytree=c(0.8,1),subsample=c(0.8,1))

xgbm4<- train(factor(costo_binario)~dias_afil_porc+ dias_afiliacion+
   TI_edad21+  TI_edad22+  TI_G_enf_totales1+ TI_estado_afiliado1+
   genero_F+ TI_tipo2+ zona_1+zona_4+zona_5+zona_9+TI_OPT_edad1+
   TI_OPT_edad4,
data=costoBinariobis,
method="xgbTree",trControl=control,
tuneGrid=xgbmgrid,verbose=FALSE)

xgbm4

plot(xgbm4)

#MEJOR CON 10
xgbmgrid<-expand.grid(
min_child_weight=20,
eta=0.01,
nrounds=1000,
max_depth=
c(5,7,9,12),gamma=0,colsample_bytree=c(0.8,1),subsample=c(0.8,1))
xgbm5<- train(factor(costo_binario)~edad_F+  edad_M+
   edad2+TI_G_enf_totales1+TI_dias_afil1+TI_tipo2+ zona_1+zona_5+
   zona_9+TI_OPT_edad2,
data=costoBinariobis,
method="xgbTree",trControl=control,
tuneGrid=xgbmgrid,verbose=FALSE)

xgbm5

plot(xgbm5)



# IMPORTANCIA DE VARIABLES

varImp(xgbm)
plot(varImp(xgbm))



# La funcion cruzadagbmbin permite plantear gradient boosting para
binarias
#



# xgboost
#MINER

medias85<-cruzadaxgbmbin(data=costoBinario, vardep="costo_binario",
listconti=c("edad_F","edad"),
listclass=c("TI_dias_afil1","genero_F", "TI_tipo2",
"TI_estado_afiliado1",
```

```
"TI_edad21",  "TI_G_enf_totales1", "TI_OPT_edad2", "TI_OPT_edad3",
   "zona_9",
"zona_2", "zona_4", "zona_5"),
grupos=4,sinicio=1234,repe=200,
min_child_weight=10,eta=0.01,nrounds=1000,max_depth=5,
gamma=0,colsample_bytree=1,subsample=1,
alpha=0,lambda=0,lambda_bias=0)

medias85$modelo="xgbm1"

#IMPORTANCIA
medias86<-cruzadaxgbmbin(data=costoBinario, vardep="costo_binario",
listconti=c("edad",  "edad2", "edad_M", "edad_F"),
listclass=c("TI_edad21", "TI_G_enf_totales1", "TI_enf_totales1",
"TI_enf_totales2"),
grupos=4,sinicio=12345,repe=200,
min_child_weight=20,eta=0.01,nrounds=2000,max_depth=5,
gamma=0,colsample_bytree=1,subsample=1,
alpha=0,lambda=0,lambda_bias=0)

medias86$modelo="xgbm2"

#ALEATORIA 1
medias87<-cruzadaxgbmbin(data=costoBinario, vardep="costo_binario",
listconti=c("dias_afiliacion", "dias_afil_porc"),
listclass=c("TI_edad21","TI_G_enf_totales1","TI_dias_afil1",
"TI_dias_afil2","TI_estado_afiliado1","genero_F","TI_tipo2",
"zona_1","zona_4","zona_5","zona_9","TI_OPT_edad1","TI_OPT_edad4"),
grupos=4,sinicio=12345,repe=200,
min_child_weight=30,eta=0.015,nrounds=1000,max_depth=5,
gamma=0,colsample_bytree=1,subsample=1,
alpha=0,lambda=0,lambda_bias=0)

medias87$modelo="xgbm3"

#ALEATORIA 2
medias88<-cruzadaxgbmbin(data=costoBinario, vardep="costo_binario",
listconti=c("dias_afiliacion", "dias_afil_porc"),
listclass=c("TI_edad21","TI_edad22","TI_G_enf_totales1","TI_estado_afi
liado1",
"genero_F","TI_tipo2","zona_1","zona_4","zona_5","zona_9","TI_OPT_edad
1","TI_OPT_edad4"),
grupos=4,sinicio=12345,repe=200,
min_child_weight=20,eta=0.015,nrounds=1000,max_depth=5,
gamma=0,colsample_bytree=1,subsample=1,
alpha=0,lambda=0,lambda_bias=0)

medias88$modelo="xgbm4"

#MEJOR CON 10
medias89<-cruzadaxgbmbin(data=costoBinario, vardep="costo_binario",
listconti=c("edad_F","edad_M", "edad2"),
listclass=c("TI_G_enf_totales1", "TI_dias_afil1", "TI_tipo2", "zona_1",
   "zona_5", "zona_9", "TI_OPT_edad2"),
grupos=4,sinicio=12345,repe=200,
min_child_weight=20,eta=0.01,nrounds=1000,max_depth=5,
gamma=0,colsample_bytree=1,subsample=1,
alpha=0,lambda=0,lambda_bias=0)

medias89$modelo="xgbm5"
```

```
#SELECCION MINER CAMBIANDO SEMILLA
medias92<-cruzadaxgbmbin(data=costoBinario, vardep="costo_binario",
listconti=c("edad_F","edad"),
listclass=c("TI_dias_afil1","genero_F", "TI_tipo2",
"TI_estado_afiliado1",
"TI_edad21",  "TI_G_enf_totales1", "TI_OPT_edad2",  "TI_OPT_edad3",
  "zona_9",
"zona_2", "zona_4", "zona_5"),
grupos=4,sinicio=12346,repe=200,
min_child_weight=10,eta=0.01,nrounds=1000,max_depth=5,
gamma=0,colsample_bytree=1,subsample=1,
alpha=0,lambda=0,lambda_bias=0)

medias92$modelo="xgbm6"

#IMPORTANCIA CAMBIANDO SEMILLA
medias93<-cruzadaxgbmbin(data=costoBinario, vardep="costo_binario",
listconti=c("edad",  "edad2",  "edad_M", "edad_F"),
listclass=c("TI_edad21", "TI_G_enf_totales1", "TI_enf_totales1",
"TI_enf_totales2"),
grupos=4,sinicio=12347,repe=200,
min_child_weight=20,eta=0.01,nrounds=2000,max_depth=5,
gamma=0,colsample_bytree=1,subsample=1,
alpha=0,lambda=0,lambda_bias=0)

medias93$modelo="xgbm7"

#ALEATORIA 1
medias94<-cruzadaxgbmbin(data=costoBinario, vardep="costo_binario",
listconti=c("dias_afiliacion", "dias_afil_porc"),
listclass=c("TI_edad21","TI_G_enf_totales1","TI_dias_afil1",
"TI_dias_afil2","TI_estado_afiliado1","genero_F","TI_tipo2",
"zona_1","zona_4","zona_5","zona_9","TI_OPT_edad1","TI_OPT_edad4"),
grupos=4,sinicio=12347,repe=200,
min_child_weight=30,eta=0.015,nrounds=1000,max_depth=5,
gamma=0,colsample_bytree=1,subsample=1,
alpha=0,lambda=0,lambda_bias=0)

medias94$modelo="xgbm8"

#ALEATORIA 2
medias95<-cruzadaxgbmbin(data=costoBinario, vardep="costo_binario",
listconti=c("dias_afiliacion", "dias_afil_porc"),
listclass=c("TI_edad21","TI_edad22","TI_G_enf_totales1","TI_estado_afi
liado1",
"genero_F","TI_tipo2","zona_1","zona_4","zona_5","zona_9","TI_OPT_edad
1","TI_OPT_edad4"),
grupos=4,sinicio=12347,repe=200,
min_child_weight=20,eta=0.015,nrounds=1000,max_depth=5,
gamma=0,colsample_bytree=1,subsample=1,
alpha=0,lambda=0,lambda_bias=0)

medias95$modelo="xgbm9"

#MEJOR CON 10
medias96<-cruzadaxgbmbin(data=costoBinario, vardep="costo_binario",
listconti=c("edad_F","edad_M","edad2"),
listclass=c("TI_G_enf_totales1", "TI_dias_afil1","TI_tipo2",  "zona_1",
  "zona_5",
```

```
"zona_9", "TI_OPT_edad2"),
grupos=4,sinicio=12347,repe=200,
min_child_weight=20,eta=0.01,nrounds=1000,max_depth=5,
gamma=0,colsample bytree=1,subsample=1,
alpha=0,lambda=0,lambda_bias=0)

medias96$modelo="xgbm10"

# PARA REDUCIR LA VARIANZA SE PUEDE RECURRIR A SORTEAR VARIABLES
# (ESTILO RANDOMFOREST PERO ANTES DEL ARBOL) Y/O
# A SORTEAR OBSERVACIONES (BAGGING)
# EN AMBOS CASOS HAY QUE AUMENTAR EL NÃšMERO DE Ã RBOLES

#ORGANIZAR1111111111111111111111111111111111


#MEJOR CON 10 SORTEO DE VARIABLES
medias100<-cruzadaxgbmbin(data=costoBinario, vardep="costo_binario",
listconti=c("edad_F","edad_M", "edad2"),
listclass=c("TI_G_enf_totales1", "TI_dias_afil1", "TI_tipo2", "zona_1",
   "zona_5", "zona_9", "TI_OPT_edad2"),
grupos=4,sinicio=12347,repe=200,
min_child_weight=20,eta=0.01,nrounds=1000,max_depth=5,
gamma=0,colsample_bytree=0.8,subsample=1,
alpha=0,lambda=0,lambda_bias=0)

medias100$modelo="xgbm11"


#MEJOR CON 10 SORTEO VARIABLES Y REGULARIZACION
medias101<-cruzadaxgbmbin(data=costoBinario, vardep="costo_binario",
listconti=c("edad_F","edad_M", "edad2"),
listclass=c("TI_G_enf_totales1", "TI_dias_afil1", "TI_tipo2", "zona_1",
   "zona_5", "zona_9", "TI_OPT_edad2"),
grupos=4,sinicio=12347,repe=200,
min_child_weight=20,eta=0.01,nrounds=1000,max_depth=5,
gamma=0,colsample_bytree=0.8,subsample=1,
alpha=0,lambda=10,lambda_bias=0)

medias101$modelo="xgbm12"

#ALEATORIA 2
medias102<-cruzadaxgbmbin(data=costoBinario, vardep="costo_binario",
listconti=c("dias_afiliacion", "dias_afil_porc"),
listclass=c("TI_edad21","TI_edad22","TI_G_enf_totales1","TI_estado_afi
liado1",
"genero_F","TI_tipo2","zona_1","zona_4","zona_5","zona_9","TI_OPT_edad
1","TI_OPT_edad4"),
grupos=4,sinicio=12345,repe=200,
min_child_weight=20,eta=0.01,nrounds=1000,max_depth=5,
gamma=0,colsample_bytree=1,subsample=1,
alpha=0,lambda=0,lambda_bias=0)

medias102$modelo="xgbm13"


medias103<-cruzadaxgbmbin(data=costoBinario, vardep="costo_binario",
listconti=c("dias_afiliacion", "dias_afil_porc"),
listclass=c("TI_edad21","TI_edad22","TI_G_enf_totales1","TI_estado_afi
liado1",
```

```
"genero_F","TI_tipo2","zona_1","zona_4","zona_5","zona_9","TI_OPT_edad
1","TI_OPT_edad4"),
grupos=4,sinicio=12345,repe=200,
min child weight=20,eta=0.01,nrounds=1000,max_depth=5,
gamma=0,colsample_bytree=0.8,subsample=1,
alpha=0,lambda=0,lambda_bias=0)

medias103$modelo="xgbm14"



# PODEMOS PROBAR ANADIR REGULARIZACION AL MEJOR MODELO XGBM

medias104<-cruzadaxgbmbin(data=costoBinario, vardep="costo_binario",
listconti=c("dias_afiliacion", "dias_afil_porc"),
listclass=c("TI_edad21","TI_edad22","TI_G_enf_totales1","TI_estado_afi
liado1",
"genero_F","TI_tipo2","zona_1","zona_4","zona_5","zona_9","TI_OPT_edad
1","TI_OPT_edad4"),
grupos=4,sinicio=12345,repe=200,
min_child_weight=20,eta=0.01,nrounds=1000,max_depth=5,
gamma=0,colsample_bytree=1,subsample=1,
alpha=0,lambda=10,lambda_bias=0)

medias104$modelo="xgbm15"


medias105<-cruzadaxgbmbin(data=costoBinario, vardep="costo_binario",
listconti=c("dias_afiliacion", "dias_afil_porc"),
listclass=c("TI_edad21","TI_edad22","TI_G_enf_totales1","TI_estado_afi
liado1",
"genero_F","TI_tipo2","zona_1","zona_4","zona_5","zona_9","TI_OPT_edad
1","TI_OPT_edad4"),
grupos=4,sinicio=12345,repe=200,
min_child_weight=20,eta=0.01,nrounds=1000,max_depth=5,
gamma=0,colsample_bytree=0.8,subsample=1,
alpha=0,lambda=10,lambda_bias=0)

medias105$modelo="xgbm16"



medias106<-cruzadaxgbmbin(data=costoBinario, vardep="costo_binario",
listconti=c("dias_afiliacion", "dias_afil_porc"),
listclass=c("TI_edad21","TI_edad22","TI_G_enf_totales1","TI_estado_afi
liado1",
"genero_F","TI_tipo2","zona_1","zona_4","zona_5","zona_9","TI_OPT_edad
1","TI_OPT_edad4"),
grupos=4,sinicio=12345,repe=200,
min_child_weight=20,eta=0.01,nrounds=1000,max_depth=10,
gamma=0,colsample_bytree=0.8,subsample=1,
alpha=0,lambda=10,lambda_bias=0)

medias106$modelo="xgbm17"


medias107<-cruzadaxgbmbin(data=costoBinario, vardep="costo_binario",
listconti=c("dias_afiliacion", "dias_afil_porc"),
listclass=c("TI_edad21","TI_edad22","TI_G_enf_totales1","TI_estado_afi
liado1",
```

```
"genero_F","TI_tipo2","zona_1","zona_4","zona_5","zona_9","TI_OPT_edad
1","TI_OPT_edad4"),
grupos=4,sinicio=12345,repe=200,
min child weight=20,eta=0.01,nrounds=1000,max_depth=10,
gamma=1,colsample_bytree=0.8,subsample=1,
alpha=0,lambda=10,lambda_bias=0)

medias107$modelo="xgbm18"

union1<-rbind(medias85,medias86,medias87, medias88, medias89,
medias92, medias93, medias94, medias95, medias96, medias100,
medias101, medias102, medias103, medias104, medias105, medias106)
uni<-union1
uni$modelo <- with(uni,
reorder(modelo,tasa, mean))

par(cex.axis=1.2,las=2)
boxplot(data=uni,tasa~modelo,col="gray", main="Tasa de fallos",
xlab="", ylab="")
boxplot(data=union1,auc~modelo,main="AUC", col="gray",xlab="",
ylab="")
```

Máquinas de soporte vectorial

```
# *************************
# TUNEADO SVM BINARIA
# *************************

setwd("C:/")
source ("cruzada SVM binaria lineal.R")
source ("cruzada SVM binaria polinomial.R")
source ("cruzada SVM binaria RBF.R")

library(caret)
library(dummies)
library (dplyr)
library (pROC)
library(kernlab)

#  SVM LINEAL: SOLO PARAMETRO C
SVMgrid<-expand.grid(C=c(0.001,0.01,0.05,0.1,0.2,0.5,1,2,5,7,10,12,20,
40,60,80,100))

SVMgrid<-expand.grid(C=c(0.01,0.02, 0.03, 0.04,0.05,0.06))
control<-trainControl(method = "cv",number=4,savePredictions = "all")

#MINER
SVM<- train(data=costoBinariobis,factor(costo_binario)~edad_F+  edad+
  TI_dias_afil1+genero_F+ TI_tipo2+ TI_estado_afiliado1+  TI_edad21+
  TI_G_enf_totales1+ TI_OPT_edad2+  TI_OPT_edad3+  zona_9+zona_2+
  zona_4+zona_5,
method="svmLinear",trControl=control,
tuneGrid=SVMgrid,verbose=FALSE)

SVM$results
plot(SVM$results$C,SVM$results$Accuracy)

#IMPORTANCIA
SVM1<- train(data=costoBinariobis,factor(costo_binario)~edad+ edad2+
  edad_M+edad_F+TI_edad21+  TI_G_enf_totales1+ TI_enf_totales1+
  TI_enf_totales2,
```

```
method="svmLinear",trControl=control,
tuneGrid=SVMgrid,verbose=FALSE)

SVM1$results
plot(SVM1$results$C,SVM1$results$Accuracy)

#ALEATORIA 1
SVM2<-
train(data=costoBinariobis,factor(costo_binario)~dias_afiliacion+
  dias_afil_porc+  TI_edad21+  TI_G_enf_totales1+ TI_dias_afil1+
  TI_dias_afil2+TI_estado_afiliado1+  genero_F+ TI_tipo2+ zona_1+
  zona_4+zona_5+zona_9+TI_OPT_edad1+  TI_OPT_edad4,
method="svmLinear",trControl=control,
tuneGrid=SVMgrid,verbose=FALSE)

SVM2$results
plot(SVM2$results$C,SVM2$results$Accuracy)


#ALEATORIA 2
SVM3<-
train(data=costoBinariobis,factor(costo_binario)~dias_afil_porc+
  dias_afiliacion+ TI_edad21+  TI_edad22+  TI_G_enf_totales1+
  TI_estado_afiliado1+  genero_F+ TI_tipo2+ zona_1+zona_4+zona_5+
  zona_9+TI_OPT_edad1+  TI_OPT_edad4,
method="svmLinear",trControl=control,
tuneGrid=SVMgrid,verbose=FALSE)

SVM3$results
plot(SVM3$results$C,SVM3$results$Accuracy)

#MEJOR CON 10
SVM4<- train(data=costoBinariobis,factor(costo_binario)~edad_F+
  edad_M+edad2+ TI_G_enf_totales1+ TI_dias_afil1+TI_tipo2+ zona_1+
  zona_5+zona_9+TI_OPT_edad2,
method="svmLinear",trControl=control,
tuneGrid=SVMgrid,verbose=FALSE)

SVM4$results
plot(SVM4$results$C,SVM4$results$Accuracy)

#  SVM Polinomial: PARAMETROS C, degree, scale

SVMgrid<-expand.grid(C=c(0.001,0.01,0.05,0.1,1,5,10),
degree=c(2,3),scale=c(0.1,0.5,2,5))

SVMgrid<-expand.grid(C=c(0.001,0.01,0.02, 0.03, 0.05),
degree=c(2,3),scale=c(0.1,0.5,1,2,5))


control<-trainControl(method = "cv",
number=4,savePredictions = "all")

#MINER
SVM5<- train(data=costoBinariobis,factor(costo_binario)~edad_F+ edad+
  TI_dias_afil1+genero_F+ TI_tipo2+ TI_estado_afiliado1+  TI_edad21+
  TI_G_enf_totales1+ TI_OPT_edad2+  TI_OPT_edad3+  zona_9+zona_2+
  zona_4+zona_5,
method="svmPoly",trControl=control,
tuneGrid=SVMgrid,verbose=FALSE)
```

```
SVM5

SVM5$results
plot(SVM5$results$C,SVM5$results$Accuracy)

#IMPORTANCIA
SVM6<- train(data=costoBinariobis,factor(costo_binario)~edad+ edad2+
  edad_M+edad_F+TI_edad21+  TI_G_enf_totales1+ TI_enf_totales1+
  TI_enf_totales2,
method="svmPoly",trControl=control,
tuneGrid=SVMgrid,verbose=FALSE)

SVM6$results
plot(SVM6$results$C,SVM6$results$Accuracy)

#ALEATORIA 1
SVM7<-
train(data=costoBinariobis,factor(costo_binario)~dias_afiliacion+
  dias_afil_porc+  TI_edad21+  TI_G_enf_totales1+ TI_dias_afil1+
  TI_dias_afil2+TI_estado_afiliado1+  genero_F+ TI_tipo2+ zona_1+
  zona_4+zona_5+zona_9+TI_OPT_edad1+  TI_OPT_edad4,
method="svmPoly",trControl=control,
tuneGrid=SVMgrid,verbose=FALSE)

SVM7$results
plot(SVM7$results$C,SVM7$results$Accuracy)
SVM7

#ALEATORIA 2
SVM8<-
train(data=costoBinariobis,factor(costo_binario)~dias_afil_porc+
  dias_afiliacion+ TI_edad21+  TI_edad22+  TI_G_enf_totales1+
  TI_estado_afiliado1+  genero_F+ TI_tipo2+ zona_1+zona_4+zona_5+
  zona_9+TI_OPT_edad1+  TI_OPT_edad4,
method="svmPoly",trControl=control,
tuneGrid=SVMgrid,verbose=FALSE)

SVM8$results
plot(SVM8$results$C,SVM8$results$Accuracy)

#MEJOR CON 10
SVM9<- train(data=costoBinariobis,factor(costo_binario)~edad_F+
  edad_M+edad2+ TI_G_enf_totales1+ TI_dias_afil1+TI_tipo2+ zona_1+
  zona_5+zona_9+TI_OPT_edad2,
method="svmPoly",trControl=control,
tuneGrid=SVMgrid,verbose=FALSE)

SVM9$results
plot(SVM9$results$C,SVM9$results$Accuracy)
SVM9

#RADIAL IMPORTANCIA

SVMgrid<-expand.grid(C=c(0.01,0.05,0.1,0.2,0.5,1,2,5,10,30),
sigma=c(0.01,0.05,0.1,0.2,0.5,1,2,5,10,30))

control<-trainControl(method = "cv",
number=4,savePredictions = "all")


#MINER
```

```
SVM10<- train(data=costoBinariobis,factor(costo_binario)~edad_F+
  edad+  TI_dias_afil1+genero_F+ TI_tipo2+ TI_estado_afiliado1+
  TI_edad21+  TI_G_enf_totales1+ TI_OPT_edad2+ TI_OPT_edad3+ zona_9+
  zona 2+zona 4+zona 5,
method="svmRadial",trControl=control,
tuneGrid=SVMgrid,verbose=FALSE)

SVM10

SVM10$results
plot(SVM10$results$C,SVM10$results$Accuracy)

#IMPORTANCIA
SVM11<- train(data=costoBinariobis,factor(costo_binario)~edad+  edad2+
  edad_M+edad_F+TI_edad21+  TI_G_enf_totales1+ TI_enf_totales1+
  TI_enf_totales2,
method="svmRadial",trControl=control,
tuneGrid=SVMgrid,verbose=FALSE)

SVM11$results
plot(SVM11$results$C,SVM11$results$Accuracy)

SVM11

#ALEATORIA 1
SVM12<-
train(data=costoBinariobis,factor(costo_binario)~dias_afiliacion+
  dias_afil_porc+  TI_edad21+  TI_G_enf_totales1+ TI_dias_afil1+
  TI_dias_afil2+TI_estado_afiliado1+  genero_F+ TI_tipo2+ zona_1+
  zona_4+zona_5+zona_9+TI_OPT_edad1+  TI_OPT_edad4,
method="svmRadial",trControl=control,
tuneGrid=SVMgrid,verbose=FALSE)

SVM12$results
plot(SVM12$results$C,SVM12$results$Accuracy)

SVM12

#ALEATORIA 2
SVM13<-
train(data=costoBinariobis,factor(costo_binario)~dias_afil_porc+
  dias_afiliacion+ TI_edad21+  TI_edad22+  TI_G_enf_totales1+
  TI_estado_afiliado1+  genero_F+ TI_tipo2+ zona_1+zona_4+zona_5+
  zona_9+TI_OPT_edad1+  TI_OPT_edad4,
method="svmRadial",trControl=control,
tuneGrid=SVMgrid,verbose=FALSE)

SVM13$results
plot(SVM13$results$C,SVM13$results$Accuracy)

SVM13

#MEJOR CON 10
SVM14<- train(data=costoBinariobis,factor(costo binario)~edad F+
  edad_M+edad2+ TI_G_enf_totales1+ TI_dias_afil1+TI_tipo2+ zona_1+
  zona_5+zona_9+TI_OPT_edad2,
method="svmRadial",trControl=control,
tuneGrid=SVMgrid,verbose=FALSE)

SVM14$results
plot(SVM14$results$C,SVM14$results$Accuracy)
```

SVM14

```
#VALIDACION CRUZADA
#MINER
medias110<-cruzadaSVMbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("edad_F","edad"),
listclass=c("TI_dias_afil1","genero_F", "TI_tipo2",
   "TI_estado_afiliado1",  "TI_edad21",   "TI_G_enf_totales1",
   "TI_OPT_edad2",  "TI_OPT_edad3",  "zona_9","zona_2","zona_4",
   "zona_5"),
grupos=4,sinicio=1234,repe=200,
C=0.05)

medias110$modelo="SVML1"


#IMPORTANCIA
medias111<-cruzadaSVMbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("edad",   "edad2",  "edad_M","edad_F"),
listclass=c("TI_edad21",  "TI_G_enf_totales1",  "TI_enf_totales1",
   "TI_enf_totales2"),
grupos=4,sinicio=1234,repe=200,
C=0.01)

medias111$modelo="SVML2"

#ALEATORIA 1
medias112<-cruzadaSVMbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("dias_afiliacion", "dias_afil_porc"),
listclass=c("TI_edad21",  "TI_G_enf_totales1",  "TI_dias_afil1",
   "TI_dias_afil2", "TI_estado_afiliado1",  "genero_F", "TI_tipo2",
   "zona_1", "zona_4", "zona_5", "zona_9", "TI_OPT_edad1",
   "TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=200,
C=0.001)

medias112$modelo="SVML3"


#ALEATORIA 2
medias113<-cruzadaSVMbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("dias_afiliacion", "dias_afil_porc"),
listclass=c("TI_edad21",  "TI_edad22",   "TI_G_enf_totales1",
   "TI_estado_afiliado1",  "genero_F", "TI_tipo2", "zona_1", "zona_4",
   "zona_5", "zona_9", "TI_OPT_edad1",  "TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=200,
C=0.01)

medias113$modelo="SVML4"



#MEJOR CON 10
medias114<-cruzadaSVMbin(data=costoBinariobis, vardep="costo_binario",
listconti=c("edad_F","edad_M","edad2"),
listclass=c(  "TI_G_enf_totales1",  "TI_dias_afil1", "TI_tipo2",
   "zona_1", "zona_5", "zona_9", "TI_OPT_edad2"),
grupos=4,sinicio=1234,repe=200,
C=5)
```

```
medias114$modelo="SVML5"


union1<-rbind(medias110, medias111, medias112, medias113, medias114)
uni<-union1
uni$modelo <- with(uni,
reorder(modelo,tasa, mean))

par(cex.axis=1.2,las=2)
boxplot(data=uni,tasa~modelo,col="gray", main="Tasa de fallos",
xlab="", ylab="")
boxplot(data=union1,auc~modelo,main="AUC", col="gray",xlab="",
ylab="")

meansT <-apply(medias114[,2],2,mean)
meansT

meansA <-apply(medias114[,3],2,mean)
meansA
#POLYNOMIAL

# medias52<-cruzadaSVMbinPoly(data=costoBinariobis,
vardep="costo_binario",
#
listconti=c("edad","edad2","edad_F","edad_M","dias_afiliacion"),
#
listclass=c("TI_G_enf_totales1","TI_enf_totales1","TI_edad21","TI_enf_
totales2","TI_OPT_edad24"),
#                              grupos=4,sinicio=1234,repe=200,
#                              C=0.04,degree=3,scale=5)
#
# medias52$modelo="SVMPoly"



#RADIAL
#MINER
medias120<-cruzadaSVMbinRBF(data=costoBinariobis,
vardep="costo_binario",
listconti=c("edad_F","edad"),
listclass=c("TI_dias_afil1","genero_F", "TI_tipo2",
"TI_estado_afiliado1",
"TI_edad21",  "TI_G_enf_totales1",  "TI_OPT_edad2",  "TI_OPT_edad3",
  "zona_9",
"zona_2", "zona_4", "zona_5"),
grupos=4,sinicio=1234,repe=200,
C=0.5,sigma=0.1)

medias120$modelo="SVMRBF1"

#IMPORTANCIA
medias121<-cruzadaSVMbinRBF(data=costoBinariobis,
vardep="costo binario",
listconti=c("edad",  "edad2",  "edad_M", "edad_F"),
listclass=c("TI_edad21", "TI_G_enf_totales1", "TI_enf_totales1",
"TI_enf_totales2"),
grupos=4,sinicio=1234,repe=200,
C=0.1,sigma=5)

medias121$modelo="SVMRBF2"
```

```
#ALEATORIA 1
medias122<-cruzadaSVMbinRBF(data=costoBinariobis,
vardep="costo binario",
listconti=c("dias_afiliacion", "dias_afil_porc"),
listclass=c("TI_edad21","TI_G_enf_totales1","TI_dias_afil1",
"TI_dias_afil2","TI_estado_afiliado1","genero_F","TI_tipo2",
"zona_1","zona_4","zona_5","zona_9","TI_OPT_edad1","TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=200,
C=0.1,sigma=0.2)

medias122$modelo="SVMRBF3"

#ALEATORIA 2
medias123<-cruzadaSVMbinRBF(data=costoBinariobis,
vardep="costo_binario",
listconti=c("dias_afiliacion", "dias_afil_porc"),
listclass=c("TI_edad21","TI_edad22","TI_G_enf_totales1","TI_estado_afi
liado1",
"genero_F","TI_tipo2","zona_1","zona_4","zona_5","zona_9","TI_OPT_edad
1","TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=200,
C=0.5,sigma=0.2)

medias123$modelo="SVMRBF4"


#MEJOR CON 10
medias124<-cruzadaSVMbinRBF(data=costoBinariobis,
vardep="costo_binario",
listconti=c("edad_F","edad_M", "edad2"),
listclass=c("TI_G_enf_totales1", "TI_dias_afil1","TI_tipo2", "zona_1",
   "zona_5",
"zona_9", "TI_OPT_edad2"),
grupos=4,sinicio=1234,repe=200,
C=30,sigma=0.2)

medias124$modelo="SVMRBF5"

# SVM RBF
union1<-rbind(medias120, medias121, medias122, medias123, medias124)
uni<-union1
uni$modelo <- with(uni,
reorder(modelo,tasa, mean))

par(cex.axis=1,las=2)
boxplot(data=uni,tasa~modelo,col="gray", main="Tasa de fallos",
xlab="", ylab="")
boxplot(data=union1,auc~modelo,main="AUC", col="gray",xlab="",
ylab="")

meansT <-apply(medias124[,2],2,mean)
meansT

meansA <-apply(medias124[,3],2,mean)
meansA


# EVALUACION
union1<-rbind(medias9, medias17, medias29, medias44, medias74,
medias105, medias114, medias124)
```

```
uni<-union1
uni$modelo <- with(uni,
reorder(modelo,tasa, mean))

par(cex.axis=1,las=2)
boxplot(data=uni,tasa~modelo,col="gray", main="Tasa de fallos",
xlab="", ylab="")
boxplot(data=union1,auc~modelo,main="AUC", col="gray",xlab="",
ylab="")
```

Ensamblado

```
# PRUEBAS DE ENSAMBLADO


# ************************************
# IMPORTANTE: AQUÃ  HAY QUE DECIDIR ANTES LOS PARÃ METROS A UTILIZAR
# EN CADA ALGORITMO, NO VALE GRID
# Importante, la dependiente en letras Yes, No
# PreparaciÃ³n de archivo, variables y CV.
# Esto se cambia para cada archivo.
# Necesario haber cambiado la var dep a Yes,No.
# ************************************

# LEER LAS CRUZADAS DE ENSAMBLADO, SON LIGERAMENTE DIFERENTES
# A LAS UTILIZADAS ANTERIORMENTE AUNQUE SE LLAMAN IGUAL
library(MASS)
library(dplyr)
library (caret)
setwd("C:/")
source("cruzadas ensamblado binaria fuente.R")
source ("cruzada SVM binaria lineal.R")
# source ("cruzada SVM binaria polinomial.R")
source ("cruzada SVM binaria RBF.R")


set.seed(12345)

archivo<-costoBinariobis

vardep<-"costo_binario"
listconti<-c("dias_afiliacion", "dias_afil_porc")
listclass<-
c("TI_edad21","TI_edad22","TI_G_enf_totales1","TI_estado_afiliado1",
"genero_F","TI_tipo2","zona_1","zona_4","zona_5","zona_9","TI_OPT_edad
1","TI_OPT_edad4")
grupos<-4
sinicio<-1234
repe<-50


# APLICACION CRUZADAS PARA ENSAMBLAR

medias130<-cruzadalogistica(data=archivo,
vardep=vardep,listconti=c("edad_F", "edad_M", "edad2"),
listclass=c("TI_G_enf_totales1", "TI_dias_afil1","TI_tipo2", "zona_1",
  "zona_5",
"zona_9", "TI_OPT_edad2"),grupos=grupos,sinicio=sinicio,repe=repe)

medias130bis<-as.data.frame(medias130[1])
medias130bis$modelo<-"Logistica"
predi130<-as.data.frame(medias130[2])
```

```
predi130$logi<-predi130$Yes

medias131<-cruzadaavnnetbin(data=archivo,
vardep=vardep,listconti=c("edad F", "edad M", "edad2"),
listclass=c("TI_G_enf_totales1",  "TI_dias_afil1","TI_tipo2",  "zona_1",
   "zona_5",
"zona_9", "TI_OPT_edad2"),grupos=grupos,sinicio=sinicio,repe=repe,
size=c(3),decay=c(0.001),repeticiones=30,itera=100)

medias131bis<-as.data.frame(medias131[1])
medias131bis$modelo<-"avnnet"
predi131<-as.data.frame(medias131[2])
predi131$avnnet<-predi131$Yes


medias132<-cruzadarfbin(data=archivo,
vardep=vardep,listconti=c("dias_afiliacion", "dias_afil_porc"),
listclass=c("TI_edad21","TI_edad22","TI_G_enf_totales1","TI_estado_afi
liado1",
"genero_F","TI_tipo2","zona_1","zona_4","zona_5","zona_9","TI_OPT_edad
1","TI_OPT_edad4"),grupos=grupos,sinicio=sinicio,repe=repe,
mtry=14,ntree=1000,nodesize=30,replace=TRUE,sampsize=2000)


medias132bis<-as.data.frame(medias132[1])
medias132bis$modelo<-"rf"
predi132<-as.data.frame(medias132[2])
predi132$rf<-predi132$Yes

medias133<-cruzadagbmbin(data=archivo,
vardep=vardep,listconti=c("dias_afiliacion", "dias_afil_porc"),
listclass=c("TI_edad21","TI_edad22","TI_G_enf_totales1","TI_estado_afi
liado1",
"genero_F","TI_tipo2","zona_1","zona_4","zona_5","zona_9","TI_OPT_edad
1","TI_OPT_edad4"),
grupos=grupos,sinicio=sinicio,repe=repe,
n.minobsinnode=30,shrinkage=0.01,n.trees=2000,interaction.depth=2)

medias133bis<-as.data.frame(medias133[1])
medias133bis$modelo<-"gbm"
predi133<-as.data.frame(medias133[2])
predi133$gbm<-predi133$Yes

medias134<-cruzadaxgbmbin(data=archivo,
vardep=vardep,listconti=c("dias_afiliacion", "dias_afil_porc"),
listclass=c("TI_edad21","TI_edad22","TI_G_enf_totales1","TI_estado_afi
liado1",
"genero_F","TI_tipo2","zona_1","zona_4","zona_5","zona_9","TI_OPT_edad
1","TI_OPT_edad4"),
grupos=grupos,sinicio=sinicio,repe=repe,
min_child_weight=20,eta=0.01,nrounds=1000,max_depth=5,
gamma=0,colsample_bytree=0.8,subsample=1,
alpha=0,lambda=10,lambda_bias=0)



medias134bis<-as.data.frame(medias134[1])
medias134bis$modelo<-"xgbm"
predi134<-as.data.frame(medias134[2])
predi134$xgbm<-predi134$Yes
```

```
medias135<-cruzadaSVMbin(data=archivo,
vardep=vardep,listconti=c("edad_F", "edad_M", "edad2"),
listclass=c("TI G enf totales1", "TI dias afil1", "TI_tipo2", "zona_1",
   "zona_5", "zona_9", "TI_OPT_edad2"),grupos=grupos,
sinicio=sinicio,repe=repe,C=5)


medias135bis<-as.data.frame(medias135[1])
medias135bis$modelo<-"svmLinear"
predi135<-as.data.frame(medias135[2])
predi135$svmLinear<-predi135$Yes


# medias7<-cruzadaSVMbinPoly(data=archivo,
#    vardep=vardep,listconti=listconti,
#    listclass=listclass,grupos=grupos,sinicio=sinicio,repe=repe,
#  C=0.01,degree=2,scale=0.1)
#
# medias7bis<-as.data.frame(medias7[1])
# medias7bis$modelo<-"svmPoly"
# predi7<-as.data.frame(medias7[2])
# predi7$svmPoly<-predi7$Yes

medias136<-cruzadaSVMbinRBF(data=archivo,
vardep=vardep,listconti=c("edad_F", "edad_M", "edad2"),
listclass=c("TI_G_enf_totales1", "TI_dias_afil1", "TI_tipo2", "zona_1",
   "zona_5", "zona_9", "TI_OPT_edad2"),grupos=grupos,
sinicio=sinicio,repe=repe,
C=30,sigma=0.2)

medias136bis<-as.data.frame(medias136[1])
medias136bis$modelo<-"svmRadial"
predi136<-as.data.frame(medias136[2])
predi136$svmRadial<-predi136$Yes

union1<-rbind(medias130bis,medias131bis,
medias132bis,medias133bis,medias134bis, medias135bis,medias136bis)

par(cex.axis=0.8)
boxplot(data=union1,tasa~modelo,main="TASA", col="gray")
boxplot(data=union1,auc~modelo,main="AUC", col="gray")

# CONSTRUCCION DE TODOS LOS ENSAMBLADOS
# SE UTILIZARAN LOS ARCHIVOS SURGIDOS DE LAS FUNCIONES LLAMADOS
predi1,...

unipredi<-cbind(predi130,predi131,predi132,predi133,predi134)

# Esto es para eliminar columnas duplicadas
unipredi<- unipredi[, !duplicated(colnames(unipredi))]

# Construccion de ensamblados

unipredi$predi79<-(unipredi$logi+unipredi$avnnet)/2
unipredi$predi80<-(unipredi$logi+unipredi$rf)/2
unipredi$predi81<-(unipredi$logi+unipredi$gbm)/2
unipredi$predi82<-(unipredi$logi+unipredi$xgbm)/2
unipredi$predi83<-(unipredi$logi+unipredi$svmLinear)/2

unipredi$predi85<-(unipredi$logi+unipredi$svmRadial)/2
```

```
unipredi$predi86<-(unipredi$avnnet+unipredi$rf)/2
unipredi$predi87<-(unipredi$avnnet+unipredi$gbm)/2
unipredi$predi18<-(unipredi$avnnet+unipredi$xgbm)/2
unipredi$predi19<-(unipredi$avnnet+unipredi$svmLinear)/2

unipredi$predi21<-(unipredi$avnnet+unipredi$svmRadial)/2
unipredi$predi22<-(unipredi$rf+unipredi$gbm)/2
unipredi$predi23<-(unipredi$rf+unipredi$xgbm)/2
unipredi$predi24<-(unipredi$rf+unipredi$svmLinear)/2

unipredi$predi26<-(unipredi$rf+unipredi$svmRadial)/2
unipredi$predi27<-(unipredi$gbm+unipredi$xgbm)/2
unipredi$predi28<-(unipredi$gbm+unipredi$svmLinear)/2

unipredi$predi30<-(unipredi$gbm+unipredi$svmRadial)/2

unipredi$predi31<-(unipredi$logi+unipredi$avnnet+unipredi$rf)/3
unipredi$predi32<-(unipredi$logi+unipredi$avnnet+unipredi$gbm)/3
unipredi$predi33<-(unipredi$logi+unipredi$avnnet+unipredi$xgbm)/3
unipredi$predi34<-(unipredi$logi+unipredi$avnnet+unipredi$svmLinear)/3

unipredi$predi36<-(unipredi$logi+unipredi$avnnet+unipredi$svmRadial)/3
unipredi$predi37<-(unipredi$logi+unipredi$rf+unipredi$gbm)/3
unipredi$predi38<-(unipredi$logi+unipredi$rf+unipredi$xgbm)/3
unipredi$predi39<-(unipredi$logi+unipredi$rf+unipredi$svmLinear)/3

unipredi$predi41<-(unipredi$logi+unipredi$rf+unipredi$svmRadial)/3
unipredi$predi42<-(unipredi$logi+unipredi$gbm+unipredi$xgbm)/3
unipredi$predi43<-(unipredi$logi+unipredi$gbm+unipredi$xgbm)/3
unipredi$predi44<-(unipredi$logi+unipredi$gbm+unipredi$svmLinear)/3

unipredi$predi46<-(unipredi$logi+unipredi$gbm+unipredi$svmRadial)/3
unipredi$predi47<-(unipredi$logi+unipredi$xgbm+unipredi$svmLinear)/3

unipredi$predi49<-(unipredi$logi+unipredi$xgbm+unipredi$svmRadial)/3

unipredi$predi50<-(unipredi$rf+unipredi$gbm+unipredi$svmLinear)/3

unipredi$predi52<-(unipredi$rf+unipredi$gbm+unipredi$svmRadial)/3

unipredi$predi53<-(unipredi$rf+unipredi$xgbm+unipredi$svmLinear)/3

unipredi$predi55<-(unipredi$rf+unipredi$xgbm+unipredi$svmRadial)/3

unipredi$predi56<-(unipredi$rf+unipredi$avnnet+unipredi$gbm)/3
unipredi$predi57<-(unipredi$rf+unipredi$avnnet+unipredi$xgbm)/3
unipredi$predi58<-(unipredi$rf+unipredi$avnnet+unipredi$svmLinear)/3

unipredi$predi60<-(unipredi$rf+unipredi$avnnet+unipredi$svmRadial)/3

unipredi$predi61<-(unipredi$avnnet+unipredi$gbm+unipredi$svmLinear)/3

unipredi$predi63<-(unipredi$avnnet+unipredi$gbm+unipredi$svmRadial)/3

unipredi$predi64<-
(unipredi$logi+unipredi$rf+unipredi$gbm+unipredi$avnnet)/4
unipredi$predi65<-
(unipredi$logi+unipredi$rf+unipredi$xgbm+unipredi$avnnet)/4
unipredi$predi66<-
(unipredi$logi+unipredi$rf+unipredi$xgbm+unipredi$avnnet)/4
```

```
unipredi$predi67<-
(unipredi$logi+unipredi$rf+unipredi$xgbm+unipredi$avnnet+unipredi$svmL
inear)/5

unipredi$predi69<-
(unipredi$logi+unipredi$rf+unipredi$xgbm+unipredi$avnnet+unipredi$svmR
adial)/5


# Listado de modelos a considerar, cambiar al gusto

dput(names(unipredi))

listado<-c( "logi", "avnnet", "rf",
"gbm", "xgbm", "predi79", "predi80",
"predi81", "predi82", "predi86", "predi87", "predi18", "predi22",
"predi23", "predi27", "predi31", "predi32", "predi33", "predi37",
"predi38", "predi42", "predi43", "predi56", "predi57", "predi64",
"predi65", "predi66")



# Cambio a Yes, No, todas las predicciones

for (prediccion in listado)
{
unipredi[,prediccion]<-ifelse(unipredi[,prediccion]>0.5,"Yes","No")
}

# Defino funcion tasafallos

tasafallos<-function(x,y) {
confu<-confusionMatrix(x,y)
tasa<-confu[[3]][1]
return(tasa)
}

# Se obtiene el numero de repeticiones CV y se calculan las medias por
repe en
# el data frame medias0

repeticiones<-nlevels(factor(unipredi$Rep))
unipredi$Rep<-as.factor(unipredi$Rep)
unipredi$Rep<-as.numeric(unipredi$Rep)


medias0<-data.frame(c())
for (prediccion in listado)
{
for (repe in 1:repeticiones)
{
paso <- unipredi[(unipredi$Rep==repe),]
pre<-factor(paso[,prediccion])
obs<-paso[,c("obs")]
tasa=1-tasafallos(pre,obs)
t<-as.data.frame(tasa)
t$modelo<-prediccion
medias0<-rbind(medias0,t)
}
}
```

```
# Finalmente boxplot (solo he calculado tasa fallos)

par(cex.axis=0.9,las=2)
boxplot(data=medias0,tasa~modelo, col="gray")


# PRESENTACION TABLA MEDIAS

tablamedias<-medias0 %>%
group_by(modelo) %>%
summarize(tasa=mean(tasa))

tablamedias<-tablamedias[order(tablamedias$tasa),]


# ORDENACIÃ‘N DEL FACTOR MODELO POR LAS MEDIAS EN TASA
# PARA EL GRAFICO

medias0$modelo <- with(medias0,
reorder(modelo,tasa, mean))
par(cex.axis=0.9,las=2)
boxplot(data=medias0,tasa~modelo,col="gray", xlab="", ylab="",
main="tasa de fallos")



# Se pueden escoger listas pero el factor hay que pasarlo a character
# para que no salgan en el boxplot todos los niveles del factor

listadobis<-c("logi", "avnnet",
"predi9", "predi11", "predi32", "predi33")


medias0$modelo<-as.character(medias0$modelo)

mediasver<-medias0[medias0$modelo %in% listadobis,]


mediasver$modelo <- with(mediasver,
reorder(modelo,tasa, mean))

par(cex.axis=0.9,las=2)
boxplot(data=mediasver,tasa~modelo,col="gray")


#GRAFICOS PARA OBSERVAR PREDICCIONES DE DIFERENTES ALGORITMOS

unipredi<-
cbind(predi11,predi12,predi13,predi14,predi15,predi16,predi17)
# Esto es para eliminar columnas duplicadas
unipredi<- unipredi[, !duplicated(colnames(unipredi))]

# AÃ±adir ensamblados

unipredi$predi9<-(unipredi$logi+unipredi$avnnet)/2
unipredi$predi11<-(unipredi$logi+unipredi$gbm)/2
unipredi$predi32<-(unipredi$logi+unipredi$avnnet+unipredi$gbm)/3
unipredi$predi33<-(unipredi$logi+unipredi$avnnet+unipredi$xgbm)/3

# Correlaciones entre predicciones de cada algoritmo individual
```

```
unigraf<-unipredi[unipredi$Rep=="1",]

solos<-c("logi", "avnnet",
"rf","gbm",  "xgbm")

mat<-unigraf[,solos]
matrizcorr<-cor(mat)
matrizcorr


library(corrplot)
corrplot(matrizcorr, type = "upper", order = "hclust",
tl.col ="black", tl.srt = 45,cl.lim=c(0.7,1),is.corr=FALSE
```

Resultados primera parte

```
#regresion logistica
library(glmnet)

datalog <-databis

datalog$costo_binario<-ifelse(datalog$costo_binario=="Yes",1,0)

set.seed(2784)
partitionIndex <- createDataPartition(datalog$costo_binario, p=1,
list=FALSE)
data_train <- datalog[partitionIndex,]
data_test <- datalog[-partitionIndex,]


modeloganador<-glm(costo_binario~edad_F+edad_M+
   edad2+TI_G_enf_totales1+TI_dias_afil1+TI_tipo2+ zona_1+zona_5+
   zona_9+TI_OPT_edad2,
data_train[,1:38],family=binomial)

modeloganador$fitted.values

prediccionbinaria<-data.frame(modeloganador$data)
prediccionbinaria$prediccion=data.frame(modeloganador$fitted.values)
write.csv(prediccionbinaria, file="prediccionbinaria.csv")
predict(modeloganador,costoBinario)
modeloganador[,prediccion]
modeloganador[,prediccion]<-
ifelse(modeloganador[,prediccion]>0.5,"Yes","No")
```

Modelo de dos partes Variable objetivo continua (Parte 2).
Redes – Regresión

```
library(sas7bdat)
library(caret)
library(dplyr)
library(dummies)

#Cargar el archivo
costomedio<-read.sas7bdat("C:/Users/camil/Desktop/TFM Diana Octubre
Continua/CostoMedio.sas7bdat")


dput(names(costomedio))
data<-costomedio
#
# c("id_afiliado", "costo_medio", "edad", "dias_afiliacion",
"dias_afil_porc",
```

```
#   "menos1", "dialisis", "oncologia_adultos",
"reumatologia_colageno",
#   "VIH", "zona_1", "zona_2", "zona_4", "zona_5", "zona_9",
"zona 10",
#   "zona_11", "edad2", "edad_F", "edad_M", "TI_edad21", "TI_edad22",
#   "TI_G_enf_totales1", "TI_dias_afil1", "TI_dias_afil2",
"TI_OPT_edad21",
#   "TI_OPT_edad22", "TI_OPT_edad24", "TI_enf_totales1",
"TI_enf_totales2",
#   "TI_estado_afiliado1", "genero_F", "TI_tipo1", "TI_tipo2",
"TI_OPT_edad1",
#   "TI_OPT_edad2", "TI_OPT_edad3", "TI_OPT_edad4")

#definicion de variables
listconti<-c("edad", "dias_afiliacion", "dias_afil_porc","edad2",
"edad_F","edad_M")
listclass<-c("menos1", "dialisis", "oncologia_adultos",
"reumatologia_colageno",
"VIH", "zona_1", "zona_2", "zona_4", "zona_5", "zona_9", "zona_10",
"zona_11", "TI_edad21", "TI_edad22",
"TI_G_enf_totales1", "TI_dias_afil1", "TI_dias_afil2",
"TI_OPT_edad21",
"TI_OPT_edad22", "TI_OPT_edad24", "TI_enf_totales1",
"TI_enf_totales2",
"TI_estado_afiliado1", "genero_F", "TI_tipo1", "TI_tipo2",
"TI_OPT_edad1",
"TI_OPT_edad2", "TI_OPT_edad3", "TI_OPT_edad4")
#las categoricas pasadas a dummies
databis<-costomedio

#estandarizar las variables continuas

# Calculo medias y dtipica de datos y estandarizo (solo las continuas)

means <-apply(databis[,listconti],2,mean)
sds<-sapply(databis[,listconti],sd)

# Estandarizo solo las continuas y uno con las categoricas

datacon<-scale(databis[,listconti], center = means, scale = sds)
numerocont<-which(colnames(databis)%in%listconti)
databis<-cbind(datacon,databis[,-numerocont,drop=FALSE ])

cruzadaavnnet<-
function(data=data,vardep="vardep",
listconti="listconti",listclass="listclass",
grupos=4,sinicio=1234,repe=5,
size=c(5),decay=c(0.01),repeticiones=5,itera=100)

{
library(caret)
#permite comparar modelos, hacer validacion cruzada repetida, training
test,  cambiar # nodos, no se puede
#usar early stopping, no se puede cambiar funcion de activacion ni
algoritmo

library(dplyr)
library(dummies)

# PreparaciÃ³n del archivo
```

```
# b)pasar las categÃ³ricas a dummies

# if  (listclass!=c(""))
# {
#  databis<-data[,c(vardep,listconti,listclass)]
#  databis<- dummy.data.frame(databis, listclass, sep = ".")
# }  else   {
#  databis<-data[,c(vardep,listconti)]
# }
#
# c)estandarizar las variables continuas

# Calculo medias y dtipica de datos y estandarizo (solo las continuas)

# means <-apply(databis[,listconti],2,mean)
# sds<-sapply(databis[,listconti],sd)
#
# Estandarizo solo las continuas y uno con las categoricas
#
# datacon<-scale(databis[,listconti], center = means, scale = sds)
# numerocont<-which(colnames(databis)%in%listconti)
# databis<-cbind(datacon,databis[,-numerocont,drop=FALSE ])

formu<-formula(paste(vardep,"~.",sep=""))

# Preparo caret

set.seed(sinicio)
control<-trainControl(method = "repeatedcv",
number=grupos,repeats=repe,
savePredictions = "all")

# Aplico caret y construyo modelo

avnnetgrid <-  expand.grid(size=size,decay=decay,bag=FALSE)

avnnet<- train(formu,data=databis,
method="avNNet",linout = TRUE,maxit=itera,repeats=repeticiones,
trControl=control,tuneGrid=avnnetgrid)

print(avnnet$results)

preditest<-avnnet$pred

preditest$prueba<-strsplit(preditest$Resample,"[.]")
preditest$Fold <- sapply(preditest$prueba, "[", 1)
preditest$Rep <- sapply(preditest$prueba, "[", 2)
preditest$prueba<-NULL

preditest$error<-(preditest$pred-preditest$obs)^2

medias<-preditest %>%
group_by(Rep) %>%
summarize(error=mean(error))

return(medias)

}

cruzadalin<-
function(data=data,vardep="vardep",
```

```
listconti="listconti",listclass="listclass",
grupos=4,sinicio=1234,repe=5,
size=c(5),decay=c(0.01),repeticiones=5,itera=100)

{
library(caret)
library(dplyr)
library(dummies)

# PreparaciÃ³n del archivo

# b)pasar las categÃ³ricas a dummies

# if  (listclass!=c(""))
# {
#  databis<-data[,c(vardep,listconti,listclass)]
#  databis<- dummy.data.frame(databis, listclass, sep = ".")
# }  else   {
#  databis<-data[,c(vardep,listconti)]
# }
#
# c)estandarizar las variables continuas

# Calculo medias y dtipica de datos y estandarizo (solo las continuas)

means <-apply(databis[,listconti],2,mean)
sds<-sapply(databis[,listconti],sd)

# Estandarizo solo las continuas y uno con las categoricas

datacon<-scale(databis[,listconti], center = means, scale = sds)
numerocont<-which(colnames(databis)%in%listconti)
databis<-cbind(datacon,databis[,-numerocont,drop=FALSE ])

formu<-formula(paste(vardep,"~.",sep=""))

# Preparo caret

set.seed(sinicio)
control<-trainControl(method = "repeatedcv",
number=grupos,repeats=repe,
savePredictions = "all")

# Aplico caret y construyo modelo

lineal<- train(formu,data=databis,
method="lm",trControl=control)

print(lineal$results)

preditest<-lineal$pred

preditest$prueba<-strsplit(preditest$Resample,"[.]")
preditest$Fold <- sapply(preditest$prueba, "[", 1)
preditest$Rep <- sapply(preditest$prueba, "[", 2)
preditest$prueba<-NULL

preditest$error<-(preditest$pred-preditest$obs)^2

medias<-preditest %>%
group_by(Rep) %>%
```

```
summarize(error=mean(error))

return(medias)

}

data<-databis

## Ejemplo de utilizacion cruzada AVNNET

#regresiones
#MINER
medias201<-cruzadalin(data=data,
vardep="costo_medio",listconti=c("edad2",  "edad","edad_F"),
listclass=c("oncologia_adultos",  "TI_G_enf_totales1",  "TI_OPT_edad22",
   "TI_OPT_edad4",  "TI_tipo2", "zona_2", "dialisis"),
grupos=10,sinicio=1234,repe=100)

medias201$modelo="LR1"

#IMPORTANCIA
medias202<-cruzadalin(data=data,
vardep="costo_medio",listconti=c("edad","edad2"),
listclass=c("TI_G_enf_totales1",  "TI_enf_totales1",  "TI_enf_totales2",
   "VIH",  "dialisis",  "oncologia_adultos"),
grupos=10,sinicio=12345,repe=100)

medias202$modelo="LR2"

#ALEATORIA 1
medias203<-cruzadalin(data=data,
vardep="costo_medio",listconti=c("dias_afiliacion",  "edad2"),
listclass=c("menos1","dialisis",  "oncologia_adultos",
   "reumatologia_colageno",   "VIH",  "TI_enf_totales1",
   "TI_enf_totales2",  "TI_OPT_edad3"),
grupos=10,sinicio=1234,repe=100)

medias203$modelo="LR3"


#ALEATORIA 2
medias204<-cruzadalin(data=data,
vardep="costo_medio",listconti=c("dias_afiliacion",  "edad2"),
listclass=c("menos1","dialisis",  "oncologia_adultos",
   "reumatologia_colageno",   "TI_enf_totales1",  "TI_enf_totales2",
   "TI_OPT_edad4"),
grupos=10,sinicio=1234,repe=100)

medias204$modelo="LR4"


union1<-rbind(medias201,medias202,medias203,medias204)

par(cex.axis=1)
boxplot(data=union1,error~modelo,  col="gray")
par(mar = rep(2,  4))



# ***************************
# TUNING CON CARET
```

```
# ***************************

set.seed(12345)

# Training test repetido
control<-trainControl(method = "LGOCV",p=0.7,number=10,savePredictions
= "all")


# EN LO SUCESIVO APLICAMOS VALIDACIÃ“N CRUZADA REPETIDA

set.seed(12345)
# ValidaciÃ³n cruzada repetida
control<-trainControl(method =
"repeatedcv",number=10,repeats=20,savePredictions = "all")


# ************************************************************
# avNNet: parÃ¡metros
# Number of Hidden Units (size, numeric)
# Weight Decay (decay, numeric)
# Bagging (bag, logical)
# ************************************************************
avnnetgrid <-
expand.grid(size=c(3,6,9,12,15),decay=c(0.01,0.1,0.001),bag=FALSE)

#seleccion miner
redavnnet<- train(costo_medio~edad2+  edad+  edad_F+oncologia_adultos+
   TI_G_enf_totales1+ TI_OPT_edad22+TI_OPT_edad4+  TI_tipo2+ zona_2+
   dialisis,
data=databis,
method="avNNet",linout =
TRUE,maxit=100,trControl=control,repeats=5,tuneGrid=avnnetgrid)

redavnnet

# size  15 decay 0.1

avnnetgrid <-
expand.grid(size=c(3,6,9,12,15,16,18),decay=c(0.01,0.1,0.001),bag=FALS
E)

#seleccion importancia
redavnnet2<- train(costo_medio~edad+  edad2+ TI_G_enf_totales1+
   TI_enf_totales1+ TI_enf_totales2+ VIH+ dialisis+ oncologia_adultos,
data=databis,
method="avNNet",linout =
TRUE,maxit=100,trControl=control,repeats=3,tuneGrid=avnnetgrid)

redavnnet2

# size 3  decay 0.1

avnnetgrid <-
expand.grid(size=c(3,6,9,11,13,15),decay=c(0.01,0.1,0.001),bag=FALSE)

#seleccion Aleatoria1
redavnnet3<- train(costo_medio~dias_afiliacion+edad2+ menos1+dialisis+
   oncologia_adultos+ reumatologia_colageno+  VIH+ TI_enf_totales1+
   TI_enf_totales2+ TI_OPT_edad3,
data=databis,
```

```
method="avNNet",linout =
TRUE,maxit=100,trControl=control,repeats=5,tuneGrid=avnnetgrid)

redavnnet3

# size 3 decay 0.1

#seleccion Aleatoria2
redavnnet4<- train(costo_medio~dias_afiliacion+edad2+ menos1+dialisis+
  oncologia_adultos+ reumatologia_colageno+  TI_enf_totales1+
  TI_enf_totales2+ TI_OPT_edad4,
data=databis,
method="avNNet",linout =
TRUE,maxit=100,trControl=control,repeats=5,tuneGrid=avnnetgrid)

redavnnet4

# size  decay

#MINER
medias205<-cruzadaavnnet(data=databis,
vardep="costo_medio",listconti=c("edad2",  "edad","edad_F"),
listclass=c("oncologia_adultos", "TI_G_enf_totales1", "TI_OPT_edad22",
  "TI_OPT_edad4",  "TI_tipo2", "zona_2", "dialisis"),
grupos=10,sinicio=1234,repe=50,
size=c(15),decay=c(0.1),repeticiones=50,itera=50)

medias205$modelo="red"

#IMPORTANCIA
medias206<-cruzadaavnnet(data=databis,
vardep="costo_medio",listconti=c("edad","edad2"),
listclass=c("TI_G_enf_totales1", "TI_enf_totales1", "TI_enf_totales2",
  "VIH", "dialisis", "oncologia_adultos"),
grupos=10,sinicio=1234,repe=50,
size=c(3),decay=c(0.1),repeticiones=50,itera=50)

medias206$modelo="red2"

#ALEATORIA 1
medias207<-cruzadaavnnet(data=databis,
vardep="costo_medio",listconti=c("dias_afiliacion", "edad2"),
listclass=c("menos1","dialisis", "oncologia_adultos",
  "reumatologia_colageno",  "VIH", "TI_enf_totales1",
  "TI_enf_totales2", "TI_OPT_edad3"),
grupos=10,sinicio=1234,repe=50,
size=c(3),decay=c(0.1),repeticiones=50,itera=50)

medias207$modelo="red3"

#ALEATORIA 2
medias208<-cruzadaavnnet(data=databis,
vardep="costo_medio",listconti=c("dias_afiliacion", "edad2"),
listclass=c("menos1","dialisis", "oncologia adultos",
  "reumatologia_colageno",  "TI_enf_totales1", "TI_enf_totales2",
  "TI_OPT_edad4"),
grupos=10,sinicio=1234,repe=50,
size=c(3),decay=c(0.1),repeticiones=50,itera=50)

medias208$modelo="red4"
```

```
## Union de las medias y presentacion del boxplot

union1<-rbind(medias205,medias206,medias207,medias208)

par(cex.axis=1)
boxplot(data=union1,error~modelo, col="gray", main="ERROR")
```

Bagging – Random forest

```
library(caret)
library(lattice)
library (pROC)
setwd("C:/Users/camil/Desktop/TFM Diana/Arboles R")
source("cruzada rf continua.R")
library(randomForest)
library(dplyr)

# VARIABLE CONTINUA
# La funcion cruzadarf permite plantear bagging PARA VDEP CONTINUA
# (para bagging hay que poner mtry=numero de variables independientes)
# No se puede plotear oob

#MINER
rfbis<-randomForest(costo_medio~edad2+  edad+  edad_F+
  oncologia_adultos+ TI_G_enf_totales1+ TI_OPT_edad22+TI_OPT_edad4+
  TI_tipo2+ zona_2+dialisis,
data=databis,
mtry=10,ntree=2000,sampsize=1000,nodesize=30,replace=TRUE)

plot(rfbis$mse)
rfbis

#IMPORTANCIA
rfbis2<-randomForest(costo_medio~edad+  edad2+ TI_G_enf_totales1+
  TI_enf_totales1+ TI_enf_totales2+ VIH+ dialisis+ oncologia_adultos,
data=databis,
mtry=9,ntree=2000,sampsize=1000,nodesize=30,replace=TRUE)

plot(rfbis2$mse)

#ALEATORIA 1
rfbis3<-randomForest(costo_medio~dias_afiliacion+ edad2+ menos1+
  dialisis+ oncologia_adultos+ reumatologia_colageno+  VIH+
  TI_enf_totales1+ TI_enf_totales2+ TI_OPT_edad3,
data=databis,
mtry=10,ntree=2000,sampsize=1000,nodesize=30,replace=TRUE)

plot(rfbis3$mse)

#ALEATORIA 2
rfbis4<-randomForest(costo_medio~dias_afiliacion+ edad2+ menos1+
  dialisis+ oncologia_adultos+ reumatologia_colageno+  TI_enf_totales1+
  TI_enf_totales2+ TI_OPT_edad4,
data=databis,
mtry=9,ntree=2000,sampsize=1000,nodesize=30,replace=TRUE)

plot(rfbis4$mse)



#MINER
medias210<-cruzadarf(data=data,
```

```
vardep="costo_medio",listconti=c("edad2",  "edad","edad_F"),
listclass=c("oncologia_adultos", "TI_G_enf_totales1",  "TI_OPT_edad22",
   "TI_OPT_edad4",  "TI_tipo2", "zona_2", "dialisis"),
grupos=4,sinicio=1234,repe=20,
nodesize=30,replace=TRUE,ntree=400,sampsize =1000,mtry=10)

medias210$modelo="bag1"


medias211<-cruzadarf(data=data,
vardep="costo_medio",listconti=c("edad2",  "edad","edad_F"),
listclass=c("oncologia_adultos", "TI_G_enf_totales1",  "TI_OPT_edad22",
   "TI_OPT_edad4",  "TI_tipo2", "zona_2", "dialisis"),
grupos=4,sinicio=1234,repe=20,
nodesize=30,replace=TRUE,ntree=400,sampsize =2000,mtry=10)

medias211$modelo="bag2"

medias212<-cruzadarf(data=data,
vardep="costo_medio",listconti=c("edad2",  "edad","edad_F"),
listclass=c("oncologia_adultos", "TI_G_enf_totales1",  "TI_OPT_edad22",
   "TI_OPT_edad4",  "TI_tipo2", "zona_2", "dialisis"),
grupos=4,sinicio=1234,repe=20,
nodesize=30,replace=TRUE,ntree=400,sampsize =3000,mtry=10)

medias212$modelo="bag3"


#IMPORTANCIA
medias213<-cruzadarf(data=data,
vardep="costo_medio",listconti=c("edad","edad2"),
listclass=c("TI_G_enf_totales1",  "TI_enf_totales1", "TI_enf_totales2",
   "VIH",  "dialisis", "oncologia_adultos"),
grupos=4,sinicio=1234,repe=30,
nodesize=30,replace=TRUE,ntree=500,sampsize=1000,mtry=8)

medias213$modelo="bag4"

medias214<-cruzadarf(data=data,
vardep="costo_medio",listconti=c("edad","edad2"),
listclass=c("TI_G_enf_totales1",  "TI_enf_totales1", "TI_enf_totales2",
   "VIH",  "dialisis", "oncologia_adultos"),
grupos=4,sinicio=1234,repe=30,
nodesize=30,replace=TRUE,ntree=500,sampsize=2000,mtry=8)

medias214$modelo="bag5"

medias215<-cruzadarf(data=data,
vardep="costo_medio",listconti=c("edad","edad2"),
listclass=c("TI_G_enf_totales1",  "TI_enf_totales1", "TI_enf_totales2",
   "VIH",  "dialisis", "oncologia_adultos"),
grupos=4,sinicio=1234,repe=30,
nodesize=30,replace=TRUE,ntree=500,sampsize=3000,mtry=8)

medias215$modelo="bag6"

#ALEATORIA 1
medias216<-cruzadarf(data=data,
vardep="costo_medio",listconti=c("dias_afiliacion",  "edad2"),
listclass=c("menos1","dialisis",  "oncologia_adultos",
   "reumatologia_colageno",   "VIH",
```

```
"TI_enf_totales1", "TI_enf_totales2", "TI_OPT_edad3"),
grupos=4,sinicio=1234,repe=30,
nodesize=30,replace=TRUE,ntree=1000,sampsize=1000,mtry=10)

medias216$modelo="bag7"

medias217<-cruzadarf(data=data,
vardep="costo_medio",listconti=c("dias_afiliacion", "edad2"),
listclass=c("menos1","dialisis", "oncologia_adultos",
   "reumatologia_colageno",  "VIH",
"TI_enf_totales1", "TI_enf_totales2", "TI_OPT_edad3"),
grupos=4,sinicio=1234,repe=30,
nodesize=30,replace=TRUE,ntree=1000,sampsize=2000,mtry=10)

medias217$modelo="bag8"

medias218<-cruzadarf(data=data,
vardep="costo_medio",listconti=c("dias_afiliacion", "edad2"),
listclass=c("menos1","dialisis", "oncologia_adultos",
   "reumatologia_colageno",  "VIH",
"TI_enf_totales1", "TI_enf_totales2", "TI_OPT_edad3"),
grupos=4,sinicio=1234,repe=30,
nodesize=30,replace=TRUE,ntree=1000,sampsize=3000,mtry=10)

medias218$modelo="bag9"

#ALEATORIA 2
medias219<-cruzadarf(data=data,
vardep="costo_medio",listconti=c("dias_afiliacion", "edad2"),
listclass=c("menos1","dialisis", "oncologia_adultos",
   "reumatologia_colageno",  "TI_enf_totales1",
"TI_enf_totales2", "TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=30,
nodesize=30,replace=TRUE,ntree=1000,sampsize=1000,mtry=9)

medias219$modelo="bag10"

medias220<-cruzadarf(data=data,
vardep="costo_medio",listconti=c("dias_afiliacion", "edad2"),
listclass=c("menos1","dialisis", "oncologia_adultos",
   "reumatologia_colageno",  "TI_enf_totales1",
"TI_enf_totales2", "TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=30,
nodesize=30,replace=TRUE,ntree=1000,sampsize=2000,mtry=9)

medias220$modelo="bag11"

medias221<-cruzadarf(data=data,
vardep="costo_medio",listconti=c("dias_afiliacion", "edad2"),
listclass=c("menos1","dialisis", "oncologia_adultos",
   "reumatologia_colageno",  "TI_enf_totales1",
"TI_enf_totales2", "TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=30,
nodesize=30,replace=TRUE,ntree=1000,sampsize=3000,mtry=9)

medias221$modelo="bag12"

#bagging
union1<-rbind(medias210,medias211,medias212, medias213, medias214,
medias215, medias216, medias217, medias218, medias219, medias220,
medias221)
```

```
uni<-union1
uni$modelo <- with(uni, reorder(modelo, error, mean))


par(cex.axis=1.2)
boxplot(data=uni,error~modelo, col="gray", main="Error", xlab="",
ylab="")


#RANDOM FOREST

#MINER
medias225<-cruzadarf(data=data,
vardep="costo_medio",listconti=c("edad2",  "edad","edad_F"),
listclass=c("oncologia_adultos", "TI_G_enf_totales1",  "TI_OPT_edad22",
   "TI_OPT_edad4",  "TI_tipo2", "zona_2", "dialisis"),
grupos=4,sinicio=1234,repe=20,
nodesize=30,replace=TRUE,ntree=400,sampsize =1000,mtry=8)

medias225$modelo="RF1"


medias226<-cruzadarf(data=data,
vardep="costo_medio",listconti=c("edad2",  "edad","edad_F"),
listclass=c("oncologia_adultos", "TI_G_enf_totales1",  "TI_OPT_edad22",
   "TI_OPT_edad4",  "TI_tipo2", "zona_2", "dialisis"),
grupos=4,sinicio=1234,repe=20,
nodesize=30,replace=TRUE,ntree=400,sampsize =1000,mtry=6)

medias226$modelo="RF2"

medias227<-cruzadarf(data=data,
vardep="costo_medio",listconti=c("edad2",  "edad","edad_F"),
listclass=c("oncologia_adultos", "TI_G_enf_totales1",  "TI_OPT_edad22",
   "TI_OPT_edad4",  "TI_tipo2", "zona_2", "dialisis"),
grupos=4,sinicio=1234,repe=20,
nodesize=30,replace=TRUE,ntree=400,sampsize =1000,mtry=4)

medias227$modelo="RF3"


#IMPORTANCIA
medias228<-cruzadarf(data=data,
vardep="costo_medio",listconti=c("edad","edad2"),
listclass=c("TI_G_enf_totales1",  "TI_enf_totales1", "TI_enf_totales2",
   "VIH",  "dialisis", "oncologia_adultos"),
grupos=4,sinicio=1234,repe=30,
nodesize=30,replace=TRUE,ntree=500,sampsize=1000,mtry=6)

medias228$modelo="RF4"

medias229<-cruzadarf(data=data,
vardep="costo medio",listconti=c("edad","edad2"),
listclass=c("TI_G_enf_totales1",  "TI_enf_totales1", "TI_enf_totales2",
   "VIH",  "dialisis", "oncologia_adultos"),
grupos=4,sinicio=1234,repe=30,
nodesize=30,replace=TRUE,ntree=500,sampsize=1000,mtry=4)

medias229$modelo="RF5"
```

```
#ALEATORIA 1
medias230<-cruzadarf(data=data,
vardep="costo_medio",listconti=c("dias_afiliacion", "edad2"),
listclass=c("menos1","dialisis", "oncologia_adultos",
   "reumatologia_colageno",  "VIH",
"TI_enf_totales1", "TI_enf_totales2", "TI_OPT_edad3"),
grupos=4,sinicio=1234,repe=30,
nodesize=30,replace=TRUE,ntree=1000,sampsize=1000,mtry=8)

medias230$modelo="RF6"

medias231<-cruzadarf(data=data,
vardep="costo_medio",listconti=c("dias_afiliacion", "edad2"),
listclass=c("menos1","dialisis", "oncologia_adultos",
   "reumatologia_colageno",  "VIH",
"TI_enf_totales1", "TI_enf_totales2", "TI_OPT_edad3"),
grupos=4,sinicio=1234,repe=30,
nodesize=30,replace=TRUE,ntree=1000,sampsize=1000,mtry=6)

medias231$modelo="RF7"

medias232<-cruzadarf(data=data,
vardep="costo_medio",listconti=c("dias_afiliacion", "edad2"),
listclass=c("menos1","dialisis", "oncologia_adultos",
   "reumatologia_colageno",  "VIH",
"TI_enf_totales1", "TI_enf_totales2", "TI_OPT_edad3"),
grupos=4,sinicio=1234,repe=30,
nodesize=30,replace=TRUE,ntree=1000,sampsize=1000,mtry=4)

medias232$modelo="RF8"

#ALEATORIA 2
medias233<-cruzadarf(data=data,
vardep="costo_medio",listconti=c("dias_afiliacion", "edad2"),
listclass=c("menos1","dialisis", "oncologia_adultos",
   "reumatologia_colageno",  "TI_enf_totales1",
"TI_enf_totales2", "TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=30,
nodesize=30,replace=TRUE,ntree=1000,sampsize=1000,mtry=8)

medias233$modelo="RF9"


medias234<-cruzadarf(data=data,
vardep="costo_medio",listconti=c("dias_afiliacion", "edad2"),
listclass=c("menos1","dialisis", "oncologia_adultos",
   "reumatologia_colageno",  "TI_enf_totales1",
"TI_enf_totales2", "TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=30,
nodesize=30,replace=TRUE,ntree=1000,sampsize=1000,mtry=6)

medias234$modelo="RF10"

medias235<-cruzadarf(data=data,
vardep="costo_medio",listconti=c("dias_afiliacion", "edad2"),
listclass=c("menos1","dialisis", "oncologia_adultos",
   "reumatologia_colageno",  "TI_enf_totales1",
"TI_enf_totales2", "TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=30,
nodesize=30,replace=TRUE,ntree=1000,sampsize=1000,mtry=4)
```

```
medias235$modelo="RF11"

medias236<-cruzadarf(data=data,
vardep="costo medio",listconti=c("dias afiliacion", "edad2"),
listclass=c("menos1","dialisis", "oncologia_adultos",
  "reumatologia_colageno",  "VIH",
"TI_enf_totales1", "TI_enf_totales2", "TI_OPT_edad3"),
grupos=4,sinicio=1234,repe=30,
nodesize=20,replace=TRUE,ntree=1000,sampsize=1000,mtry=4)

medias236$modelo="RF12"


#rf
union1<-
rbind(medias225,medias226,medias227,medias228,medias229,medias230,
medias231, medias232, medias233, medias234, medias235,medias236)

uni<-union1
uni$modelo <-with(uni,reorder(modelo,error,mean))

par(cex.axis=1.2)
boxplot(data=uni,error~modelo,col="gray")
```

Incremento gradiente

```
library(caret)
library(dummies)
library(dplyr)
library(pROC)
source("cruzada gbm continua.R")


# VARIABLE CONTINUA

set.seed(12345)

gbmgrid<-expand.grid(shrinkage=c(0.1,0.05,0.01,0.001),
n.minobsinnode=c(10,20,30),
n.trees=c(1000,2000,3000),
interaction.depth=c(2))

control<-trainControl(method = "cv",number=4,savePredictions = "all")

#MINER
gbm<- train(costo_medio~edad2+ edad+  edad_F+oncologia_adultos+
  TI_G_enf_totales1+ TI_OPT_edad22+TI_OPT_edad4+  TI_tipo2+ zona_2+
  dialisis,
data=databis,
method="gbm",trControl=control,tuneGrid=gbmgrid,
distribution="gaussian", bag.fraction=1,verbose=FALSE)

gbm

plot(gbm)

#IMPORTANCIA
gbm2<- train(costo_medio~edad+ edad2+  TI_G_enf_totales1+
  TI_enf_totales1+ TI_enf_totales2+ VIH+ dialisis+ oncologia_adultos,
data=databis,
method="gbm",trControl=control,tuneGrid=gbmgrid,
distribution="gaussian", bag.fraction=1,verbose=FALSE)
```

```
gbm2

plot(gbm2)

#ALEATORIA 1
gbm3<- train(costo_medio~dias_afiliacion+  edad2+ menos1+dialisis+
   oncologia_adultos+ reumatologia_colageno+  VIH+ TI_enf_totales1+
   TI_enf_totales2+ TI_OPT_edad3,
data=databis,
method="gbm",trControl=control,tuneGrid=gbmgrid,
distribution="gaussian", bag.fraction=1,verbose=FALSE)

gbm3

plot(gbm3)


#ALEATORIA 2
gbm4<- train(costo_medio~dias_afiliacion+  edad2+ menos1+dialisis+
   oncologia_adultos+ reumatologia_colageno+  TI_enf_totales1+
   TI_enf_totales2+ TI_OPT_edad4,
data=databis,
method="gbm",trControl=control,tuneGrid=gbmgrid,
distribution="gaussian", bag.fraction=1,verbose=FALSE)

gbm4

plot(gbm4)

# ESTUDIO DE EARLY STOPPING
# Probamos a fijar algunos parÃ¡metros para ver como evoluciona
# en funciÃ³n de las iteraciones

#MINER
gbmgrid<-expand.grid(shrinkage=c(0.001),
n.minobsinnode=c(20),
n.trees=c(500,1000,1200,2000,5000,7000,8000,20000),
interaction.depth=c(2))

control<-trainControl(method = "cv",number=4,savePredictions = "all")

gbm5<- train(costo_medio~edad2+  edad+  edad_F+oncologia_adultos+
   TI_G_enf_totales1+ TI_OPT_edad22+TI_OPT_edad4+  TI_tipo2+ zona_2+
   dialisis,
data=databis,
method="gbm",trControl=control,tuneGrid=gbmgrid,
distribution="gaussian", bag.fraction=1,verbose=FALSE)

gbm5

plot(gbm5)


#IMPORTANCIA
gbmgrid<-expand.grid(shrinkage=c(0.001),
n.minobsinnode=c(10),
n.trees=c(500,1000,1200,2000,5000,7000,8000,20000),
interaction.depth=c(2))

control<-trainControl(method = "cv",number=4,savePredictions = "all")
```

```
gbm6<- train(costo_medio~edad+ edad2+ TI_G_enf_totales1+
   TI_enf_totales1+ TI_enf_totales2+ VIH+ dialisis+ oncologia_adultos,
data=databis,
method="gbm",trControl=control,tuneGrid=gbmgrid,
distribution="gaussian", bag.fraction=1,verbose=FALSE)

gbm6

plot(gbm6)

#ALEATORIA 1
gbmgrid<-expand.grid(shrinkage=c(0.01),
n.minobsinnode=c(30),
n.trees=c(500,1000,1200,2000,5000,7000,8000,20000),
interaction.depth=c(2))

control<-trainControl(method = "cv",number=4,savePredictions = "all")

gbm7<- train(costo_medio~dias_afiliacion+  edad2+ menos1+dialisis+
   oncologia_adultos+ reumatologia_colageno+  VIH+ TI_enf_totales1+
   TI_enf_totales2+ TI_OPT_edad3,
data=databis,
method="gbm",trControl=control,tuneGrid=gbmgrid,
distribution="gaussian", bag.fraction=1,verbose=FALSE)

gbm7

plot(gbm7)


#ALEATORIA 2
gbmgrid<-expand.grid(shrinkage=c(0.01),
n.minobsinnode=c(30),
n.trees=c(500,1000,1200,2000,5000,7000,8000,20000),
interaction.depth=c(2))

control<-trainControl(method = "cv",number=4,savePredictions = "all")

gbm8<- train(costo_medio~dias_afiliacion+  edad2+ menos1+dialisis+
   oncologia_adultos+ reumatologia_colageno+  TI_enf_totales1+
   TI_enf_totales2+ TI_OPT_edad4,
data=databis,
method="gbm",trControl=control,tuneGrid=gbmgrid,
distribution="gaussian", bag.fraction=1,verbose=FALSE)

gbm8

plot(gbm8)



data<-databis
#MINER
medias240<-cruzadagbm(data=data,
vardep="costo_medio",listconti=c("edad2",  "edad","edad_F"),
listclass=c("oncologia_adultos",  "TI_G_enf_totales1",  "TI_OPT_edad22",
   "TI_OPT_edad4",  "TI_tipo2",  "zona_2",  "dialisis"),
grupos=4,sinicio=1234,repe=50,
n.minobsinnode=20,shrinkage=0.001,n.trees=2000,interaction.depth=2)
```

```
medias240$modelo="gbm"

#IMPORTANCIA
medias241<-cruzadagbm(data=data,
vardep="costo_medio",listconti=c("edad","edad2"),
listclass=c("TI_G_enf_totales1", "TI_enf_totales1", "TI_enf_totales2",
   "VIH", "dialisis", "oncologia_adultos"),
grupos=4,sinicio=1234,repe=50,
n.minobsinnode=10,shrinkage=0.001,n.trees=2000,interaction.depth=2)

medias241$modelo="gbm2"

#ALEATORIA 1
medias242<-cruzadagbm(data=data,
vardep="costo_medio",listconti=c("dias_afiliacion", "edad2"),
listclass=c("menos1","dialisis", "oncologia_adultos",
   "reumatologia_colageno",  "VIH", "TI_enf_totales1",
   "TI_enf_totales2", "TI_OPT_edad3"),
grupos=4,sinicio=1234,repe=50,
n.minobsinnode=30,shrinkage=0.01,n.trees=1000,interaction.depth=2)

medias242$modelo="gbm3"

#ALEATORIA 2
medias243<-cruzadagbm(data=data,
vardep="costo_medio",listconti=c("dias_afiliacion", "edad2"),
listclass=c("menos1","dialisis", "oncologia_adultos",
   "reumatologia_colageno",  "TI_enf_totales1", "TI_enf_totales2",
   "TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=50,
n.minobsinnode=30,shrinkage=0.01,n.trees=1000,interaction.depth=2)

medias243$modelo="gbm4"


medias244<-cruzadagbm(data=data,
vardep="costo_medio",listconti=c("dias_afiliacion", "edad2"),
listclass=c("menos1","dialisis", "oncologia_adultos",
   "reumatologia_colageno",  "TI_enf_totales1", "TI_enf_totales2",
   "TI_OPT_edad4"),
grupos=4,sinicio=12345,repe=50,
n.minobsinnode=30,shrinkage=0.01,n.trees=1000,interaction.depth=2)

medias244$modelo="gbm5"

union1<-rbind(medias240,medias241,medias242,medias243, medias244)

uni<-union1
uni$modelo <- with(uni, reorder(modelo, error, mean))


par(cex.axis=1.2)
boxplot(data=uni,error~modelo, col="gray", main="Error", xlab="",
ylab="")
```

Xgboost

```
library(caret)
library(dplyr)
library(xgboost)
source ("cruzada xgboost continua.R")
```

```
set.seed(12345)

xgbmgrid<-expand.grid(
min child weight=c(10,20,30),
eta=c(0.01,0.015,0.025,0.05,0.1),
nrounds=c(1000,2000,5000),
max_depth=5,gamma=0,colsample_bytree=1,subsample=1)



#MINER
xgbm<- train(costo_medio~edad2+  edad+  edad_F+oncologia_adultos+
  TI_G_enf_totales1+ TI_OPT_edad22+TI_OPT_edad4+  TI_tipo2+ zona_2+
  dialisis,
data=databis,
method="xgbTree",trControl=control,
tuneGrid=xgbmgrid,verbose=FALSE)

xgbm

plot(xgbm)


#IMPORTANCIA
xgbm2<- train(costo_medio~edad+  edad2+ TI_G_enf_totales1+
  TI_enf_totales1+ TI_enf_totales2+ VIH+ dialisis+ oncologia_adultos,
data=databis,
method="xgbTree",trControl=control,
tuneGrid=xgbmgrid,verbose=FALSE)

xgbm2

plot(xgbm2)

#ALEATORIA 1
xgbm3<- train(costo_medio~dias_afiliacion+ edad2+ menos1+dialisis+
  oncologia_adultos+ reumatologia_colageno+  VIH+ TI_enf_totales1+
  TI_enf_totales2+ TI_OPT_edad3,
data=databis,
method="xgbTree",trControl=control,
tuneGrid=xgbmgrid,verbose=FALSE)

xgbm3

plot(xgbm3)

#ALEATORIA 2
xgbm4<- train(costo_medio~dias_afiliacion+ edad2+ menos1+dialisis+
  oncologia_adultos+ reumatologia_colageno+  TI_enf_totales1+
  TI_enf_totales2+ TI_OPT_edad4,
data=databis,
method="xgbTree",trControl=control,
tuneGrid=xgbmgrid,verbose=FALSE)

xgbm4

plot(xgbm4)


# ESTUDIO DE EARLY STOPPING
# Probamos a fijar algunos parÃ¡metros para ver como evoluciona
```

```
# en funciÃ³n de las iteraciones


#MINER
xgbmgrid<-expand.grid(
min_child_weight=30,
eta=0.01,
nrounds=300,
max_depth=c(5,7,9,12),gamma=c(0,0.3, 0.5,
0.7,1),colsample_bytree=c(0.8,1),subsample=c(0.8,1))

control<-trainControl(method = "cv",number=4,savePredictions = "all",
classProbs=TRUE)

xgbm9<- train(costo_medio~edad2+ edad+  edad_F+oncologia_adultos+
   TI_G_enf_totales1+ TI_OPT_edad22+TI_OPT_edad4+ TI_tipo2+ zona_2+
   dialisis,
data=databis,
method="xgbTree",trControl=control,
tuneGrid=xgbmgrid,verbose=FALSE)

xgbm9

plot(xgbm9)


xgbmgrid<-expand.grid(eta=c(0.01),
min_child_weight=c(30),
nrounds=c(300,500,1000,1500,2000),
max_depth=5,gamma=1,colsample_bytree=1,subsample=1)

set.seed(12345)
control<-trainControl(method = "cv",number=4,savePredictions = "all")


xgbm5<- train(costo_medio~edad2+ edad+  edad_F+oncologia_adultos+
   TI_G_enf_totales1+ TI_OPT_edad22+TI_OPT_edad4+ TI_tipo2+ zona_2+
   dialisis,
data=databis,
method="xgbTree",trControl=control,
tuneGrid=xgbmgrid,verbose=FALSE)

xgbm5

plot(xgbm5)


#IMPORTANCIA
xgbmgrid<-expand.grid(eta=c(0.01),
min_child_weight=c(30),
nrounds=c(300,500,1000,1500,2000),
max_depth=5,gamma=1,colsample_bytree=1,subsample=1)

set.seed(12345)
control<-trainControl(method = "cv",number=4,savePredictions = "all")



xgbm6<- train(costo_medio~edad+  edad2+ TI_G_enf_totales1+
   TI_enf_totales1+ TI_enf_totales2+ VIH+ dialisis+ oncologia_adultos,
data=databis,
```

```
method="xgbTree",trControl=control,
tuneGrid=xgbmgrid,verbose=FALSE)

xgbm6

plot(xgbm6)

xgbmgrid<-expand.grid(
min_child_weight=30,
eta=0.01,
nrounds=300,
max_depth=c(5,7,9,12),gamma=c(0,0.3, 0.5,
0.7,1),colsample_bytree=c(0.8,1),subsample=c(0.8,1))

control<-trainControl(method = "cv",number=4,savePredictions = "all",
classProbs=TRUE)
xgbm10<- train(costo_medio~edad+ edad2+ TI_G_enf_totales1+
  TI_enf_totales1+ TI_enf_totales2+ VIH+ dialisis+ oncologia_adultos,
data=databis,
method="xgbTree",trControl=control,
tuneGrid=xgbmgrid,verbose=FALSE)

xgbm10

plot(xgbm10)


#ALEATORIA 1
xgbmgrid<-expand.grid(eta=c(0.01),
min_child_weight=c(30),
nrounds=c(300,500,1000,1500,2000),
max_depth=5,gamma=1,colsample_bytree=1,subsample=1)

set.seed(12345)
control<-trainControl(method = "cv",number=4,savePredictions = "all")



xgbm7<- train(costo_medio~dias_afiliacion+ edad2+ menos1+dialisis+
  oncologia_adultos+ reumatologia_colageno+  VIH+ TI_enf_totales1+
  TI_enf_totales2+ TI_OPT_edad3,
data=databis,
method="xgbTree",trControl=control,
tuneGrid=xgbmgrid,verbose=FALSE)

xgbm7

plot(xgbm7)


xgbmgrid<-expand.grid(
min_child_weight=30,
eta=0.01,
nrounds=300,
max_depth=c(5,7,9,12),gamma=c(0,0.3, 0.5,
0.7,1),colsample_bytree=c(0.8,1),subsample=c(0.8,1))

control<-trainControl(method = "cv",number=4,savePredictions = "all",
classProbs=TRUE)
```

```
xgbm11<- train(costo_medio~dias_afiliacion+  edad2+ menos1+dialisis+
  oncologia_adultos+ reumatologia_colageno+  VIH+ TI_enf_totales1+
  TI_enf_totales2+ TI_OPT_edad3,
data=databis,
method="xgbTree",trControl=control,
tuneGrid=xgbmgrid,verbose=FALSE)

xgbm11

plot(xgbm11)


#ALEATORIA 2
xgbmgrid<-expand.grid(eta=c(0.01),
min_child_weight=c(30),
nrounds=c(300,500,1000,1500,2000),
max_depth=5,gamma=1,colsample_bytree=1,subsample=1)

set.seed(12345)
control<-trainControl(method = "cv",number=4,savePredictions = "all")



xgbm8<- train(costo_medio~dias_afiliacion+ edad2+ menos1+dialisis+
  oncologia_adultos+ reumatologia_colageno+  TI_enf_totales1+
  TI_enf_totales2+ TI_OPT_edad4,
data=databis,
method="xgbTree",trControl=control,
tuneGrid=xgbmgrid,verbose=FALSE)

xgbm8

plot(xgbm8)

xgbmgrid<-expand.grid(
min_child_weight=30,
eta=0.01,
nrounds=300,
max_depth=c(5,7,9,12),gamma=c(0,0.3, 0.5,
0.7,1),colsample_bytree=c(0.8,1),subsample=c(0.8,1))

control<-trainControl(method = "cv",number=4,savePredictions = "all",
classProbs=TRUE)
xgbm12<- train(costo_medio~dias_afiliacion+  edad2+ menos1+dialisis+
  oncologia_adultos+ reumatologia_colageno+  TI_enf_totales1+
  TI_enf_totales2+ TI_OPT_edad4,
data=databis,
method="xgbTree",trControl=control,
tuneGrid=xgbmgrid,verbose=FALSE)

xgbm12

plot(xgbm12)

# IMPORTANCIA DE VARIABLES

varImp(xgbm)
plot(varImp(xgbm))


# UTILIZACION DE LOS PARAMETROS  DE REGULARIZACION
```

```
#VALIDACION CRUZADA
#MINER
medias250<-cruzadaxgbm(data=data,
vardep="costo_medio",listconti=c("edad2",  "edad","edad_F"),
listclass=c("oncologia_adultos", "TI_G_enf_totales1",  "TI_OPT_edad22",
   "TI_OPT_edad4",  "TI_tipo2", "zona_2", "dialisis"),
grupos=4,sinicio=1234,repe=50,
min_child_weight=30,eta=0.01,nrounds=300,max_depth=5,
gamma=1,colsample_bytree=1,subsample=1, lambda=0 )

medias250$modelo="xgbm"

#IMPORTANCIA
medias251<-cruzadaxgbm(data=data,
vardep="costo_medio",listconti=c("edad","edad2"),
listclass=c("TI_G_enf_totales1", "TI_enf_totales1", "TI_enf_totales2",
   "VIH", "dialisis", "oncologia_adultos"),
grupos=4,sinicio=1234,repe=50,
min_child_weight=30,eta=0.01,nrounds=300,max_depth=5,
gamma=1,colsample_bytree=1,subsample=1, lambda=0 )

medias251$modelo="xgbm2"

medias261<-cruzadaxgbm(data=data,
vardep="costo_medio",listconti=c("edad","edad2"),
listclass=c("TI_G_enf_totales1", "TI_enf_totales1", "TI_enf_totales2",
   "VIH", "dialisis", "oncologia_adultos"),
grupos=4,sinicio=1234,repe=50,
min_child_weight=30,eta=0.01,nrounds=300,max_depth=5,
gamma=0.3,colsample_bytree=0.8,subsample=0.8, lambda=0 )

medias261$modelo="xgbm10"


#ALEATORIA 1
medias252<-cruzadaxgbm(data=data,
vardep="costo_medio",listconti=c("dias_afiliacion", "edad2"),
listclass=c("menos1","dialisis", "oncologia_adultos",
   "reumatologia_colageno",  "VIH", "TI_enf_totales1",
   "TI_enf_totales2", "TI_OPT_edad3"),
grupos=4,sinicio=1234,repe=50,
min_child_weight=30,eta=0.01,nrounds=300,max_depth=5,
gamma=1,colsample_bytree=1,subsample=1, lambda=0 )

medias252$modelo="xgbm3"

medias262<-cruzadaxgbm(data=data,
vardep="costo_medio",listconti=c("dias_afiliacion", "edad2"),
listclass=c("menos1","dialisis", "oncologia_adultos",
   "reumatologia_colageno",  "VIH", "TI_enf_totales1",
   "TI_enf_totales2", "TI_OPT_edad3"),
grupos=4,sinicio=1234,repe=50,
min_child_weight=30,eta=0.01,nrounds=300,max_depth=5,
gamma=0.3,colsample_bytree=1,subsample=0.8, lambda=0 )

medias262$modelo="xgbm11"

#ALEATORIA 2
medias253<-cruzadaxgbm(data=data,
```

```
vardep="costo_medio",listconti=c("dias_afiliacion", "edad2"),
listclass=c("menos1","dialisis", "oncologia_adultos",
   "reumatologia_colageno",  "TI_enf_totales1", "TI_enf_totales2",
   "TI OPT edad4"),
grupos=4,sinicio=1234,repe=50,
min_child_weight=30,eta=0.01,nrounds=300,max_depth=5,
gamma=1,colsample_bytree=1,subsample=1, lambda=0 )

medias253$modelo="xgbm4"

medias263<-cruzadaxgbm(data=data,
vardep="costo_medio",listconti=c("dias_afiliacion", "edad2"),
listclass=c("menos1","dialisis", "oncologia_adultos",
   "reumatologia_colageno",  "TI_enf_totales1", "TI_enf_totales2",
   "TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=50,
min_child_weight=30,eta=0.01,nrounds=300,max_depth=5,
gamma=0.3,colsample_bytree=1,subsample=0.8, lambda=0 )

medias263$modelo="xgbm12"


#ALEATORIA 2 cambiando numero minimo de observaciones
medias254<-cruzadaxgbm(data=data,
vardep="costo_medio",listconti=c("dias_afiliacion", "edad2"),
listclass=c("menos1","dialisis", "oncologia_adultos",
   "reumatologia_colageno",  "TI_enf_totales1", "TI_enf_totales2",
   "TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=50,
min_child_weight=20,eta=0.01,nrounds=300,max_depth=5,
gamma=1,colsample_bytree=1,subsample=1, lambda=0 )

medias254$modelo="xgbm5"

#sorteo de variables
medias255<-cruzadaxgbm(data=data,
vardep="costo_medio",listconti=c("dias_afiliacion", "edad2"),
listclass=c("menos1","dialisis", "oncologia_adultos",
   "reumatologia_colageno",  "TI_enf_totales1", "TI_enf_totales2",
   "TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=50,
min_child_weight=30,eta=0.01,nrounds=300,max_depth=5,
gamma=1,colsample_bytree=0.8,subsample=1, lambda=0 )

medias255$modelo="xgbm6"


#IMPORTANCIA cambiando sorteo de variables y lambda
medias256<-cruzadaxgbm(data=data,
vardep="costo_medio",listconti=c("dias_afiliacion", "edad2"),
listclass=c("menos1","dialisis", "oncologia_adultos",
   "reumatologia_colageno",  "TI_enf_totales1", "TI_enf_totales2",
   "TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=50,
min_child_weight=30,eta=0.01,nrounds=300,max_depth=5,
gamma=1,colsample_bytree=0.8,subsample=1, lambda=10 )

medias256$modelo="xgbm7"

#IMPORTANCIA cambiando sorteo de variables y obser
medias257<-cruzadaxgbm(data=data,
```

```
vardep="costo_medio",listconti=c("dias_afiliacion", "edad2"),
listclass=c("menos1","dialisis", "oncologia_adultos",
   "reumatologia_colageno",  "TI_enf_totales1", "TI_enf_totales2",
   "TI OPT edad4"),
grupos=4,sinicio=1234,repe=50,
min_child_weight=30,eta=0.01,nrounds=300,max_depth=5,
gamma=1,colsample_bytree=0.8,subsample=0.8, lambda=10 )

medias257$modelo="xgbm8"

#IMPORTANCIA cambiando sorteo de variables y obser PROFUNDIDAD 10
medias258<-cruzadaxgbm(data=data,
vardep="costo_medio",listconti=c("dias_afiliacion", "edad2"),
listclass=c("menos1","dialisis", "oncologia_adultos",
   "reumatologia_colageno",  "TI_enf_totales1", "TI_enf_totales2",
   "TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=50,
min_child_weight=30,eta=0.01,nrounds=300,max_depth=10,
gamma=1,colsample_bytree=0.8,subsample=0.8, lambda=10 )

medias258$modelo="xgbm9"

union1<-rbind(medias250,medias251,medias252,medias253, medias254,
medias255, medias256, medias257, medias258, medias261,
medias262,medias263)

uni<-union1
uni$modelo <- with(uni, reorder(modelo, error, mean))


par(cex.axis=1.2)
boxplot(data=uni,error~modelo, col="gray", main="Error", xlab="",
ylab="")
```

Máquinas de soporte vectorial

```
library(caret)
library(dplyr)
library(dummies)

source("cruzada SVM continua lineal.R")
source("cruzada SVM continua polinomial.R")
source("cruzada SVM continua RBF.R")



# *************************
# TUNEADO SVM CONTINUA
# *************************

#  SVM LINEAL: SOLO PARAMETRO C

SVMgrid<-expand.grid(C=c(0.01,0.05,0.1,0.2,0.5,1,2,5,10))

control<-trainControl(method = "cv",number=4,
savePredictions = "all")

#MINER
SVM<- train(data=databis,
costo_medio~edad2+ edad+  edad_F+oncologia_adultos+ TI_G_enf_totales1+
   TI_OPT_edad22+TI_OPT_edad4+ TI_tipo2+ zona_2+dialisis,
method="svmLinear",trControl=control,
```

```
tuneGrid=SVMgrid,verbose=FALSE)

SVM$results

SVM

#IMPORTANCIA
SVM2<- train(data=databis,
costo_medio~edad+  edad2+ TI_G_enf_totales1+ TI_enf_totales1+
  TI_enf_totales2+ VIH+ dialisis+ oncologia_adultos,
method="svmLinear",trControl=control,
tuneGrid=SVMgrid,verbose=FALSE)

SVM2$results
SVM2

#ALEATORIA 1

SVM3<- train(data=databis,
costo_medio~dias_afiliacion+ edad2+ menos1+dialisis+ oncologia_adultos+
  reumatologia_colageno+  VIH+ TI_enf_totales1+ TI_enf_totales2+
  TI_OPT_edad3,
method="svmLinear",trControl=control,
tuneGrid=SVMgrid,verbose=FALSE)

SVM3$results
SVM3

#ALEATORIA 2

SVM4<- train(data=databis,
costo_medio~dias_afiliacion+ edad2+ menos1+dialisis+ oncologia_adultos+
  reumatologia_colageno+  TI_enf_totales1+ TI_enf_totales2+
  TI_OPT_edad4,
method="svmLinear",trControl=control,
tuneGrid=SVMgrid,verbose=FALSE)

SVM4$results
SVM4

#MINER
medias265<-cruzadaSVM(data=data,
vardep="costo_medio",listconti=c("edad2",  "edad","edad_F"),
listclass=c("oncologia_adultos", "TI_G_enf_totales1", "TI_OPT_edad22",
  "TI_OPT_edad4",  "TI_tipo2", "zona_2", "dialisis"),
grupos=4,sinicio=1234,repe=50,C=0.5)

medias265$modelo="SVML1"

#IMPORTANCIA
medias266<-cruzadaSVM(data=data,
vardep="costo_medio",listconti=c("edad","edad2"),
listclass=c("TI_G_enf_totales1", "TI_enf_totales1", "TI_enf_totales2",
  "VIH", "dialisis", "oncologia adultos"),
grupos=4,sinicio=1234,repe=50,C=0.01)

medias266$modelo="SVML2"

#ALEATORIA 1
medias267<-cruzadaSVM(data=data,
vardep="costo_medio",listconti=c("dias_afiliacion", "edad2"),
```

```
listclass=c("menos1","dialisis", "oncologia_adultos",
   "reumatologia_colageno",  "VIH", "TI_enf_totales1",
   "TI_enf_totales2", "TI_OPT_edad3"),
grupos=4,sinicio=1234,repe=50,C=2)

medias267$modelo="SVML3"

#ALEATORIA 2
medias268<-cruzadaSVM(data=data,
vardep="costo_medio",listconti=c("dias_afiliacion", "edad2"),
listclass=c("menos1","dialisis", "oncologia_adultos",
   "reumatologia_colageno",  "TI_enf_totales1", "TI_enf_totales2",
   "TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=50,C=2)

medias268$modelo="SVML4"

union1<-rbind(medias265,medias266,medias267,medias268)

uni<-union1
uni$modelo <- with(uni, reorder(modelo, error, mean))


par(cex.axis=1.2)
boxplot(data=uni,error~modelo, col="gray", main="Error", xlab="",
ylab="")


#  SVM Polinomial: PARAMETROS C, degree, scale

# SVMgrid<-expand.grid(C=c(0.001,0.01,0.02, 0.03,
0.04,0.05,0.1,1,5,10,20,40),
#                      degree=c(2,3),scale=c(0.1,0.5,1,2,5))
#
#
# control<-trainControl(method = "cv",
#  number=4,savePredictions = "all")
#
#
# SVM<- train(data=databis,
#             costo_medio~dialisis+TI_G_enf_totales1+TI_enf_totales1
#
+TI_enf_totales2+VIH+reumatologia_colageno+edad+edad2+edad_M,
#             method="svmPoly",trControl=control,
#             tuneGrid=SVMgrid,verbose=FALSE)
#
# SVM
#
# SVM$results
#
# plot(SVM$results$C,SVM$results$RMSE)

#  SVM RBF: PARAMETROS C, sigma

SVMgrid<-expand.grid(C=c(0.01,0.05,0.1,0.2,0.5,1,2,5,10),
sigma=c(0.01,0.05,0.1,0.2,0.5,1,2,5,10,30))

control<-trainControl(method = "cv",
number=4,savePredictions = "all")

#MINER
```

```
SVM5<- train(data=databis,
costo_medio~edad2+ edad+  edad_F+oncologia_adultos+ TI_G_enf_totales1+
  TI_OPT_edad22+TI_OPT_edad4+ TI_tipo2+ zona_2+dialisis,
method="svmRadial",trControl=control,
tuneGrid=SVMgrid,verbose=FALSE)

SVM5

#IMPORTANCIA
SVM6<- train(data=databis,
costo_medio~edad+  edad2+ TI_G_enf_totales1+ TI_enf_totales1+
  TI_enf_totales2+ VIH+ dialisis+ oncologia_adultos,
method="svmRadial",trControl=control,
tuneGrid=SVMgrid,verbose=FALSE)

SVM6

#ALEATORIA 1
SVM7<- train(data=databis,
costo_medio~dias_afiliacion+ edad2+ menos1+dialisis+ oncologia_adultos+
  reumatologia_colageno+  VIH+ TI_enf_totales1+ TI_enf_totales2+
  TI_OPT_edad3,
method="svmRadial",trControl=control,
tuneGrid=SVMgrid,verbose=FALSE)

SVM7

#ALEATORIA 2
SVM8<- train(data=databis,
costo_medio~dias_afiliacion+ edad2+ menos1+dialisis+ oncologia_adultos+
  reumatologia_colageno+  TI_enf_totales1+ TI_enf_totales2+
  TI_OPT_edad4,
method="svmRadial",trControl=control,
tuneGrid=SVMgrid,verbose=FALSE)

SVM8


#MINER
medias270<-cruzadaSVMRBF(data=data,
vardep="costo_medio",listconti=c("edad2",  "edad","edad_F"),
listclass=c("oncologia_adultos", "TI_G_enf_totales1", "TI_OPT_edad22",
  "TI_OPT_edad4",  "TI_tipo2", "zona_2", "dialisis"),
grupos=4,sinicio=1234,repe=50,C=10,sigma=0.05)

medias270$modelo="SVMRBF"

#IMPORTANCIA
medias271<-cruzadaSVMRBF(data=data,
vardep="costo_medio",listconti=c("edad","edad2"),
listclass=c("TI_G_enf_totales1", "TI_enf_totales1", "TI_enf_totales2",
  "VIH", "dialisis", "oncologia_adultos"),
grupos=4,sinicio=1235,repe=50,C=10,sigma=0.2)

medias271$modelo="SVMRBF2"

#ALEATORIA 1
medias272<-cruzadaSVMRBF(data=data,
vardep="costo_medio",listconti=c("dias_afiliacion", "edad2"),
```

```
listclass=c("menos1","dialisis", "oncologia_adultos",
   "reumatologia_colageno",  "VIH", "TI_enf_totales1",
   "TI_enf_totales2", "TI_OPT_edad3"),
grupos=4,sinicio=12345,repe=50,C=10,sigma=0.1)

medias272$modelo="SVMRBF3"


#ALEATORIA 2
medias273<-cruzadaSVMRBF(data=data,
vardep="costo_medio",listconti=c("dias_afiliacion", "edad2"),
listclass=c("menos1","dialisis", "oncologia_adultos",
   "reumatologia_colageno",  "TI_enf_totales1", "TI_enf_totales2",
   "TI_OPT_edad4"),
grupos=4,sinicio=12345,repe=50,C=10,sigma=0.01)

medias273$modelo="SVMRBF4"

union1<-rbind(medias270,medias271, medias272, medias273)

uni<-union1
uni$modelo <- with(uni, reorder(modelo, error, mean))


par(cex.axis=1.2)
boxplot(data=uni,error~modelo, col="gray", main="Error", xlab="",
ylab="")

#EVALUACION
union1<-rbind(medias204,medias208, medias216, medias236, medias244,
medias257, medias268, medias273)

uni<-union1
uni$modelo <- with(uni, reorder(modelo, error, mean))


par(cex.axis=1.2)
boxplot(data=uni,error~modelo, col="gray", main="Error", xlab="",
ylab="")
```

Ensamblado

```
# (variable dependiente continua)
library(dplyr)
library(reshape)
library(MASS)
library(dplyr)
setwd("C:/Users/felip/Desktop/TFM R")
source("cruzadas ensamblado continuas TFM source.R")

# Por hacer una prueba rapida, comparo regresion con rf en los 3 sets
de variables.
# Lo hago con el esquema de ensamblado, pero todavÃ?a sin ensamblar
# En cada modelo pongo las variables


archivo<-databis

vardep<-"costo_medio"
listclass<-c("TI_G_enf_totales1","TI_enf_totales1","TI_enf_totales2",
"VIH", "dialisis", "oncologia_adultos")
```

```
listconti<-c("edad_M",  "edad","edad2")
grupos<-4
sinicio<-1234
repe<-50

# APLICACION CRUZADAS PARA ENSAMBLAR

medias65<-cruzadalin(data=archivo,
vardep=vardep,listconti=c("dias_afiliacion", "edad2"),
listclass=c("menos1","dialisis", "oncologia_adultos",
   "reumatologia_colageno",  "TI_enf_totales1", "TI_enf_totales2",
   "TI_OPT_edad4"),grupos=grupos,sinicio=sinicio,repe=repe)

medias65bis<-as.data.frame(medias65[1])
medias65bis$modelo<-"regresion"
predi65<-as.data.frame(medias65[2])
predi65$reg<-predi65$pred

medias66<-cruzadaavnnet(data=archivo,
vardep=vardep,listconti=c("dias_afiliacion", "edad2"),
listclass=c("menos1","dialisis", "oncologia_adultos",
   "reumatologia_colageno",  "TI_enf_totales1", "TI_enf_totales2",
   "TI_OPT_edad4"),grupos=grupos,sinicio=sinicio,repe=repe,
size=c(3),decay=c(0.1),repeticiones=5,itera=200,trace=FALSE)

medias66bis<-as.data.frame(medias66[1])
medias66bis$modelo<-"avnnet"
predi66<-as.data.frame(medias66[2])
predi66$avnnet<-predi66$pred


medias67<-cruzadarf(data=archivo,
vardep=vardep,listconti=c("dias_afiliacion", "edad2"),
listclass=c("menos1","dialisis", "oncologia_adultos",
   "reumatologia_colageno",  "VIH", "TI_enf_totales1",
   "TI_enf_totales2",
   "TI_OPT_edad3"),grupos=grupos,sinicio=sinicio,repe=repe,
mtry=4,ntree=1000,nodesize=20,sampsize=1000,replace=TRUE)


medias67bis<-as.data.frame(medias67[1])
medias67bis$modelo<-"rf"
predi67<-as.data.frame(medias67[2])
predi67$rf<-predi67$pred

medias68<-cruzadagbm(data=archivo,
vardep=vardep,listconti=c("dias_afiliacion", "edad2"),
listclass=c("menos1","dialisis", "oncologia_adultos",
   "reumatologia_colageno",  "TI_enf_totales1", "TI_enf_totales2",
   "TI_OPT_edad4"),grupos=grupos,sinicio=sinicio,repe=repe,
n.minobsinnode=30,shrinkage=0.01,n.trees=1000,interaction.depth=2)

medias68bis<-as.data.frame(medias68[1])
medias68bis$modelo<-"gbm"
predi68<-as.data.frame(medias68[2])
predi68$gbm<-predi68$pred

medias69<-cruzadaxgbm(data=archivo,
vardep=vardep,listconti=c("dias_afiliacion", "edad2"),
```

```
listclass=c("menos1","dialisis", "oncologia_adultos",
   "reumatologia_colageno",  "TI_enf_totales1", "TI_enf_totales2",
   "TI_OPT_edad4"),grupos=grupos,sinicio=sinicio,repe=repe,
min child weight=30,eta=0.01,nrounds=300,max depth=5,
gamma=1,colsample_bytree=0.8,subsample=0.8, lambda =10)


medias69bis<-as.data.frame(medias69[1])
medias69bis$modelo<-"xgbm"
predi69<-as.data.frame(medias69[2])
predi69$xgbm<-predi69$pred


medias70<-cruzadaSVMRBF(data=archivo,
vardep=vardep,listconti=c("dias_afiliacion", "edad2"),
listclass=c("menos1","dialisis", "oncologia_adultos",
   "reumatologia_colageno",  "TI_enf_totales1", "TI_enf_totales2",
   "TI_OPT_edad4"),
grupos=grupos,sinicio=sinicio,repe=repe,
C=10,sigma=0.01)

medias70bis<-as.data.frame(medias70[1])
medias70bis$modelo<-"SVM"
predi70<-as.data.frame(medias70[2])
predi70$svm<-predi70$pred

# medias71<-cruzadaSVM(data=archivo,
#                          vardep=vardep,listconti=c("edad","edad2",
   "dias_afiliacion"),
#                          listclass=c("TI_G_enf_totales1",
   "TI_enf_totales2", "genero_F","TI_tipo2",  "VIH", "dialisis",
   "menos1", "oncologia_adultos",
#                                      "reumatologia_colageno",
   "zona_2", "TI_OPT_edad4"),grupos=grupos,sinicio=sinicio,repe=repe,
#                          C=10)
#
# medias71bis<-as.data.frame(medias71[1])
# medias71bis$modelo<-"SVMLineal"
# predi71<-as.data.frame(medias71[2])
# predi71$svm<-predi71$pred
#

union1<-rbind(medias65bis,medias66bis,
medias67bis,medias68bis,medias69bis,medias70bis)

par(cex.axis=1)
boxplot(data=union1,error~modelo,col="gray")


# CONSTRUCCION DE TODOS LOS ENSAMBLADOS
# SE UTILIZARAN LOS ARCHIVOS SURGIDOS DE LAS FUNCIONES LLAMADOS
predi1,...

unipredi<-cbind(predi65,predi66,predi67,predi68, predi69, predi70)

# Esto es para eliminar columnas duplicadas
unipredi<- unipredi[, !duplicated(colnames(unipredi))]

# Construccion de ensamblados, cambiar al gusto

unipredi$predi81<-(unipredi$reg+unipredi$avnnet)/2
```

```
unipredi$predi82<-(unipredi$reg+unipredi$rf)/2
unipredi$predi83<-(unipredi$reg+unipredi$gbm)/2
unipredi$predi84<-(unipredi$reg+unipredi$xgbm)/2
unipredi$predi85<-(unipredi$reg+unipredi$svm)/2
unipredi$predi86<-(unipredi$avnnet+unipredi$rf)/2
unipredi$predi87<-(unipredi$avnnet+unipredi$gbm)/2
unipredi$predi88<-(unipredi$avnnet+unipredi$xgbm)/2
unipredi$predi89<-(unipredi$avnnet+unipredi$svm)/2
unipredi$predi90<-(unipredi$rf+unipredi$gbm)/2
unipredi$predi91<-(unipredi$rf+unipredi$xgbm)/2
unipredi$predi92<-(unipredi$rf+unipredi$svm)/2
unipredi$predi93<-(unipredi$gbm+unipredi$xgbm)/2
unipredi$predi94<-(unipredi$gbm+unipredi$svm)/2
unipredi$predi95<-(unipredi$xgbm+unipredi$svm)/2

unipredi$predi96<-(unipredi$reg+unipredi$avnnet+unipredi$rf)/3
unipredi$predi97<-(unipredi$reg+unipredi$avnnet+unipredi$gbm)/3
unipredi$predi98<-(unipredi$reg+unipredi$avnnet+unipredi$xgbm)/3
unipredi$predi99<-(unipredi$reg+unipredi$avnnet+unipredi$svm)/3

unipredi$predi100<-(unipredi$reg+unipredi$rf+unipredi$gbm)/3
unipredi$predi101<-(unipredi$reg+unipredi$rf+unipredi$xgbm)/3
unipredi$predi102<-(unipredi$reg+unipredi$rf+unipredi$svm)/3
unipredi$predi103<-(unipredi$rf+unipredi$avnnet+unipredi$gbm)/3
unipredi$predi104<-(unipredi$rf+unipredi$gbm+unipredi$xgbm)/3
unipredi$predi105<-(unipredi$rf+unipredi$gbm+unipredi$xgbm)/3
unipredi$predi106<-(unipredi$svm+unipredi$gbm+unipredi$xgbm)/3
unipredi$predi107<-(unipredi$reg+unipredi$gbm+unipredi$xgbm)/3
unipredi$predi108<-(unipredi$reg+unipredi$gbm+unipredi$svm)/3
unipredi$predi109<-(unipredi$reg+unipredi$xgbm+unipredi$svm)/3
unipredi$predi110<-
(unipredi$reg+unipredi$rf+unipredi$gbm+unipredi$avnnet)/4
unipredi$predi111<-
(unipredi$reg+unipredi$rf+unipredi$gbm+unipredi$xgbm)/4
unipredi$predi112<-
(unipredi$reg+unipredi$svm+unipredi$gbm+unipredi$xgbm)/4
unipredi$predi113<-
(unipredi$reg+unipredi$avnnet+unipredi$gbm+unipredi$xgbm)/4
unipredi$predi114<-
(unipredi$reg+unipredi$avnnet+unipredi$svm+unipredi$xgbm)/4
unipredi$predi115<-
(unipredi$reg+unipredi$rf+unipredi$gbm+unipredi$xgbm)/4

dput(names(unipredi))


listado<-c("reg", "avnnet",
"rf", "gbm", "xgbm", "svm", "predi81", "predi82", "predi83",
"predi84", "predi85", "predi86", "predi87", "predi88", "predi89",
"predi90", "predi91", "predi92", "predi93", "predi94", "predi95",
"predi96", "predi97", "predi98", "predi99", "predi100", "predi101",
"predi102", "predi103", "predi104", "predi105", "predi106",
"predi107",
"predi108", "predi109", "predi110", "predi111", "predi112",
"predi113",
"predi114", "predi115")


repeticiones<-nlevels(factor(unipredi$Rep))
unipredi$Rep<-as.factor(unipredi$Rep)
unipredi$Rep<-as.numeric(unipredi$Rep)
```

```
# Calculo el MSE para cada repeticion de validaciÃ³n cruzada

medias0<-data.frame(c())

for (prediccion in listado)
{
paso <-unipredi[,c("obs",prediccion,"Rep")]
paso$error<-(paso[,c(prediccion)]-paso$obs)^2
paso<-paso %>%
group_by(Rep) %>%
summarize(error=mean(error))
paso$modelo<-prediccion
medias0<-rbind(medias0,paso)
}
# Finalmente boxplot

par(cex.axis=0.8,las=2)
boxplot(data=medias0,outcex=0.3,error~modelo)

# PRESENTACION TABLA MEDIAS

tablamedias<-medias0 %>%
summarize(error=mean(error))

tablamedias<-tablamedias[order(tablamedias$error),]

# ORDENACIÃ‘N DEL FACTOR MODELO POR LAS MEDIAS EN ERROR
# PARA EL GRÃ FICO

medias0$modelo <- with(medias0,
reorder(modelo,error, mean))
par(cex.axis=1.2)
boxplot(data=medias0,error~modelo,col="gray",xlab="",
ylab="",main="Error")

#GRAFICOS PARA OBSERVAR PREDICCIONES DE DIFERENTES ALGORITMOS

unipredi<-cbind(predi55,predi56,predi57,predi58,predi59,predi60)
# Esto es para eliminar columnas duplicadas
unipredi<- unipredi[, !duplicated(colnames(unipredi))]

# AÃ±adir ensamblados

unipredi$predi9<-(unipredi$logi+unipredi$avnnet)/2
unipredi$predi11<-(unipredi$logi+unipredi$gbm)/2
unipredi$predi32<-(unipredi$logi+unipredi$avnnet+unipredi$gbm)/3
unipredi$predi33<-(unipredi$logi+unipredi$avnnet+unipredi$xgbm)/3

# Correlaciones entre predicciones de cada algoritmo individual

unigraf<-unipredi[unipredi$Rep=="1",]


solos<-c("reg", "avnnet",
"rf","gbm",  "xgbm", "svm")

mat<-unigraf[,solos]
matrizcorr<-cor(mat)
matrizcorr
```

```
library(corrplot)
corrplot(matrizcorr, type = "upper", order = "hclust",
tl.col ="black", tl.srt = 45,cl.lim=c(0,1),is.corr=FALSE)
```

Resultados segunda parte

```
#####MODELO DE DOS PARTES
#PARTE 2
## Partici???n de datos
set.seed(2784)
partitionIndex <- createDataPartition(costetotal$costo_medio, p=0.8,
list=FALSE)
data_train <- costomedio[partitionIndex,]
data_test <- costomedio[-partitionIndex,]

modelo3<-lm(costo_medio~dias_afiliacion+edad2+ menos1+dialisis+
   oncologia_adultos+ reumatologia_colageno+  TI_enf_totales1+
   TI_enf_totales2+ TI_OPT_edad4,
data=data_train[,1:38])
summary(modelo3)
coef(modelo3)

prediccion3<-predict(modelo3,costetotal)
prediccionesparte2<-costetotal
prediccionesparte2$prediccion<-prediccion3
write.table(prediccionesparte2,file="predicciones3.csv",
col.names=TRUE)

Rsq<-function(modelo,varObj,datos){
testpredicted<-predict(modelo, datos)
5
testReal<-datos[,varObj]
sse <- sum((testpredicted - testReal) ^ 2)
sst <- sum((testReal - mean(testReal)) ^ 2)
1 - sse/sst
}
library(caret)

Rsq(modelo3,"costo_medio",data_test)
```

Modelización variable objetivo continua coste total
Redes - Regresión

```
library(sas7bdat)
library(caret)
library(dplyr)
library(dummies)

#Cargar el archivo
costetotal<-read.sas7bdat("C:/CosteTotal.sas7bdat")

dput(names(costetotal))
data<-costetotal
#
# c("id_afiliado", "costo_medio", "edad", "dias_afiliacion",
"dias_afil_porc",
#   "menos1", "dialisis", "oncologia_adultos",
"reumatologia_colageno",
#   "VIH", "zona_1", "zona_2", "zona_4", "zona_5", "zona_9",
"zona_10",
#   "zona_11", "edad2", "edad_F", "edad_M", "TI_edad21", "TI_edad22",
```

```
#   "TI_G_enf_totales1", "TI_dias_afil1", "TI_dias_afil2",
"TI_OPT_edad21",
#   "TI_OPT_edad22", "TI_OPT_edad24", "TI_enf_totales1",
"TI enf totales2",
#   "TI_estado_afiliado1", "genero_F", "TI_tipo1", "TI_tipo2",
"TI_OPT_edad1",
#   "TI_OPT_edad2", "TI_OPT_edad3", "TI_OPT_edad4")

#definicion de variables
listconti<-c("edad", "dias_afiliacion", "dias_afil_porc","edad2",
"edad_F","edad_M")
listclass<-c("menos1", "dialisis", "oncologia_adultos",
"reumatologia_colageno",
"VIH", "zona_1", "zona_2", "zona_4", "zona_5", "zona_9", "zona_10",
"zona_11", "TI_edad21", "TI_edad22",
"TI_G_enf_totales1", "TI_dias_afil1", "TI_dias_afil2",
"TI_OPT_edad21",
"TI_OPT_edad22", "TI_OPT_edad24", "TI_enf_totales1",
"TI_enf_totales2",
"TI_estado_afiliado1", "genero_F", "TI_tipo1", "TI_tipo2",
"TI_OPT_edad1",
"TI_OPT_edad2", "TI_OPT_edad3", "TI_OPT_edad4")
#las categoricas pasadas a dummies
databis<-costetotal

#estandarizar las variables continuas

# Calculo medias y dtipica de datos y estandarizo (solo las continuas)

means <-apply(databis[,listconti],2,mean)
sds<-sapply(databis[,listconti],sd)

# Estandarizo solo las continuas y uno con las categoricas

datacon<-scale(databis[,listconti], center = means, scale = sds)
numerocont<-which(colnames(databis)%in%listconti)
databis<-cbind(datacon,databis[,-numerocont,drop=FALSE ])

cruzadaavnnet<-
function(data=data,vardep="vardep",
listconti="listconti",listclass="listclass",
grupos=4,sinicio=1234,repe=20,
size=c(5),decay=c(0.01),repeticiones=20,itera=200)

{
library(caret)
#permite comparar modelos, hacer validacion cruzada repetida, training
test,  cambiar # nodos, no se puede
#usar early stopping, no se puede cambiar funcion de activacion ni
algoritmo

library(dplyr)
library(dummies)


formu<-formula(paste(vardep,"~.",sep=""))

# Preparo caret

set.seed(sinicio)
control<-trainControl(method = "repeatedcv",
```

```
number=grupos,repeats=repe,
savePredictions = "all")

# Aplico caret y construyo modelo

avnnetgrid <-  expand.grid(size=size,decay=decay,bag=FALSE)

avnnet<- train(formu,data=databis,
method="avNNet",linout = TRUE,maxit=itera,repeats=repeticiones,
trControl=control,tuneGrid=avnnetgrid)

print(avnnet$results)

preditest<-avnnet$pred

preditest$prueba<-strsplit(preditest$Resample,"[.]")
preditest$Fold <- sapply(preditest$prueba, "[", 1)
preditest$Rep <- sapply(preditest$prueba, "[", 2)
preditest$prueba<-NULL

preditest$error<-(preditest$pred-preditest$obs)^2

medias<-preditest %>%
group_by(Rep) %>%
summarize(error=mean(error))

return(medias)}

cruzadalin<-
function(data=data,vardep="vardep",
listconti="listconti",listclass="listclass",
grupos=4,sinicio=1234,repe=5,
size=c(5),decay=c(0.01),repeticiones=5,itera=100)

{
library(caret)
library(dplyr)
library(dummies)


# Calculo medias y dtipica de datos y estandarizo (solo las continuas)

means <-apply(databis[,listconti],2,mean)
sds<-sapply(databis[,listconti],sd)

# Estandarizo solo las continuas y uno con las categoricas

datacon<-scale(databis[,listconti], center = means, scale = sds)
numerocont<-which(colnames(databis)%in%listconti)
databis<-cbind(datacon,databis[,-numerocont,drop=FALSE ])

formu<-formula(paste(vardep,"~.",sep=""))

# Preparo caret

set.seed(sinicio)
control<-trainControl(method = "repeatedcv",
number=grupos,repeats=repe,
savePredictions = "all")

# Aplico caret y construyo modelo
```

```
lineal<- train(formu,data=databis,
method="lm",trControl=control)

print(lineal$results)

preditest<-lineal$pred

preditest$prueba<-strsplit(preditest$Resample,"[.]")
preditest$Fold <- sapply(preditest$prueba, "[", 1)
preditest$Rep <- sapply(preditest$prueba, "[", 2)
preditest$prueba<-NULL

preditest$error<-(preditest$pred-preditest$obs)^2

medias<-preditest %>%
group_by(Rep) %>%
summarize(error=mean(error))

return(medias)

}

#regresiones
#MINER
medias1<-cruzadalin(data=data,
vardep="costo_medio",listconti=c("edad2", "edad"),
listclass=c("oncologia_adultos", "TI_enf_totales2",
   "reumatologia_colageno",  "dialisis","TI_G_enf_totales1"),
grupos=10,sinicio=1234,repe=200)

medias1$modelo="LR1"

#IMPORTANCIA
medias2<-cruzadalin(data=data,
vardep="costo_medio",listconti=c("edad_M", "edad","edad2"),
listclass=c("TI_G_enf_totales1", "TI_enf_totales1", "TI_enf_totales2",
   "VIH", "dialisis", "oncologia_adultos"),
grupos=10,sinicio=12345,repe=200)

medias2$modelo="LR2"

#ALEATORIA 1
medias3<-cruzadalin(data=data,
vardep="costo_medio",listconti=c("edad","edad2",  "dias_afiliacion"),
listclass=c("TI_G_enf_totales1", "TI_enf_totales2", "genero_F",
   "TI_tipo2", "VIH", "dialisis", "menos1","oncologia_adultos",
   "reumatologia_colagen", "zona_2","TI_OPT_edad4"),
grupos=10,sinicio=1234,repe=200)

medias3$modelo="LR3"


union1<-rbind(medias1,medias2,medias3)

par(cex.axis=1)
boxplot(data=union1,error~modelo)
par(mar = rep(2, 4))


# ***************************
```

```
# TUNING CON CARET
# ************************

set.seed(12345)

# Training test repetido
control<-trainControl(method = "LGOCV",p=0.7,number=10,savePredictions
= "all")


# EN LO SUCESIVO APLICAMOS VALIDACIÃ“N CRUZADA REPETIDA

set.seed(12345)
# ValidaciÃ³n cruzada repetida
control<-trainControl(method =
"repeatedcv",number=10,repeats=20,savePredictions = "all")


# ************************************************************
# avNNet: parÃ¡metros
# Number of Hidden Units (size, numeric)
# Weight Decay (decay, numeric)
# Bagging (bag, logical)
# ************************************************************
avnnetgrid <-
expand.grid(size=c(3,6,9,12,15,18,20),decay=c(0.01,0.1,0.001),bag=FALS
E)

#seleccion miner
redavnnet<-
train(costo_medio~edad2+edad+oncologia_adultos+TI_enf_totales2+reumato
logia_colageno+  dialisis+ TI_G_enf_totales1,
data=databis,
method="avNNet",linout =
TRUE,maxit=100,trControl=control,repeats=5,tuneGrid=avnnetgrid)

redavnnet

# size 9  decay 0.1
avnnetgrid <-
expand.grid(size=c(3,6,9,12,15,16),decay=c(0.01,0.1,0.001),bag=FALSE)

#seleccion importancia
redavnnet2<-
train(costo_medio~edad_M+edad+edad2+TI_G_enf_totales1+TI_enf_totales1+
TI_enf_totales2+VIH+dialisis+oncologia_adultos,
data=databis,
method="avNNet",linout =
TRUE,maxit=100,trControl=control,repeats=3,tuneGrid=avnnetgrid)

redavnnet2

# size 9  decay 0.1
avnnetgrid <-
expand.grid(size=c(3,6,9,11),decay=c(0.01,0.1,0.001),bag=FALSE)

#seleccion Aleatoria1
redavnnet3<- train(costo_medio~edad+  edad2+ dias_afiliacion+
  TI_G_enf_totales1+ TI_enf_totales2+ genero_F+ TI_tipo2+ VIH+ dialisis+
  menos1+oncologia_adultos+ reumatologia_colageno+  zona_2+
  TI_OPT_edad4,
```

```
data=databis,
method="avNNet",linout =
TRUE,maxit=100,trControl=control,repeats=5,tuneGrid=avnnetgrid)

redavnnet3

# size 9 decay 0.1

#MINER
medias5<-cruzadaavnnet(data=databis,
vardep="costo_medio",listconti=c("edad2",  "edad"),
listclass=c("oncologia_adultos", "TI_enf_totales2",
   "reumatologia_colageno",  "dialisis", "TI_G_enf_totales1"),
grupos=10,sinicio=1234,repe=200,
size=c(9),decay=c(0.1),repeticiones=50,itera=50)

medias5$modelo="red"

#IMPORTANCIA
medias6<-cruzadaavnnet(data=databis,
vardep="costo_medio",listconti=c("edad2","edad","edad_M"),
listclass=c("TI_G_enf_totales1", "TI_enf_totales1", "TI_enf_totales2",
   "VIH", "dialisis", "oncologia_adultos"),
grupos=10,sinicio=1234,repe=200,
size=c(9),decay=c(0.1),repeticiones=50,itera=50)

medias6$modelo="red2"

#ALEATORIA
medias7<-cruzadaavnnet(data=databis,
vardep="costo_medio",listconti=c("edad","edad2",  "dias_afiliacion"),
listclass=c("TI_G_enf_totales1", "TI_enf_totales2", "genero_F",
   "TI_tipo2", "VIH", "dialisis", "menos1", "oncologia_adultos",
   "reumatologia_colagen", "zona_2", "TI_OPT_edad4"),
grupos=10,sinicio=1234,repe=200,
size=c(9),decay=c(0.1),repeticiones=50,itera=50)

medias7$modelo="red3"


## Union de las medias y presentacion del boxplot

union1<-rbind(medias5,medias6,medias7)

par(cex.axis=1)
boxplot(data=union1,error~modelo, col="gray", main="ERROR")
```

Bagging – Random forest

```
library(caret)
library(lattice)
library (pROC)
setwd("C:/")
source("cruzada rf continua.R")
library(randomForest)
library(dplyr)

# VARIABLE CONTINUA
# La funcion cruzadarf permite plantear bagging PARA VDEP CONTINUA
# (para bagging hay que poner mtry=numero de variables independientes)
# No se puede plotear oob
```

```
#MINER
rfbis<-randomForest(costo_medio~edad2+  edad+  oncologia_adultos+
  TI_enf_totales2+ reumatologia_colageno+  dialisis+ TI_G_enf_totales1,
data=databis,
mtry=7,ntree=2000,sampsize=1000,nodesize=30,replace=TRUE)

plot(rfbis$mse)

#IMPORTANCIA
rfbis2<-randomForest(costo_medio~edad_M+edad+  edad2+
  TI_G_enf_totales1+ TI_enf_totales1+ TI_enf_totales2+ VIH+ dialisis+
  oncologia_adultos,
data=databis,
mtry=9,ntree=2000,sampsize=1000,nodesize=30,replace=TRUE)

plot(rfbis2$mse)

#ALEATORIA
rfbis3<-randomForest(costo_medio~edad+  edad2+ dias_afiliacion+
  TI_G_enf_totales1+ TI_enf_totales2+ genero_F+ TI_tipo2+ VIH+ dialisis+
  menos1+oncologia_adultos+ reumatologia_colageno+  zona_2+
  TI_OPT_edad4,
data=databis,
mtry=14,ntree=2000,sampsize=1000,nodesize=30,replace=TRUE)

plot(rfbis3$mse)

#MINER
medias10<-cruzadarf(data=data,
vardep="costo_medio",listconti=c("edad2",  "edad"),
listclass=c("oncologia_adultos",
  "TI_enf_totales2","reumatologia_colageno","dialisis",
"TI_G_enf_totales1"),
grupos=4,sinicio=1234,repe=20,
nodesize=30,replace=TRUE,ntree=1500,sampsize =1000,mtry=7)

medias10$modelo="bag1"

#IMPORTANCIA
medias11<-cruzadarf(data=data,
vardep="costo_medio",listconti=c("edad_M", "edad","edad2"),
listclass=c("TI_G_enf_totales1",  "TI_enf_totales1","TI_enf_totales2",
"VIH",  "dialisis", "oncologia_adultos"),
grupos=4,sinicio=1234,repe=30,
nodesize=30,replace=TRUE,ntree=1200,sampsize=1000,mtry=9)

medias11$modelo="bag2"

#ALEATORIA
medias12<-cruzadarf(data=data,
vardep="costo_medio",listconti=c("edad","edad2",  "dias_afiliacion"),
listclass=c("TI_G_enf_totales1",  "TI_enf_totales2",  "genero_F",
"TI_tipo2",  "VIH",  "dialisis",  "menos1",  "oncologia_adultos",
"reumatologia colagen",  "zona 2",  "TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=30,
nodesize=30,replace=TRUE,ntree=1000,sampsize=1000,mtry=14)

medias12$modelo="bag3"

#MINER
medias13<-cruzadarf(data=data,
```

```
vardep="costo_medio",listconti=c("edad2",  "edad"),
listclass=c("oncologia_adultos",
   "TI_enf_totales2","reumatologia_colageno","dialisis",
"TI G enf totales1"),
grupos=4,sinicio=1234,repe=20,
nodesize=30,replace=TRUE,ntree=1500,sampsize =2000,mtry=7)

medias13$modelo="bag4"

#IMPORTANCIA
medias14<-cruzadarf(data=data,
vardep="costo_medio",listconti=c("edad_M", "edad","edad2"),
listclass=c("TI_G_enf_totales1", "TI_enf_totales1","TI_enf_totales2",
"VIH", "dialisis", "oncologia_adultos"),
grupos=4,sinicio=1234,repe=30,
nodesize=30,replace=TRUE,ntree=1200,sampsize=2000,mtry=9)

medias14$modelo="bag5"

#ALEATORIA
medias15<-cruzadarf(data=data,
vardep="costo_medio",listconti=c("edad","edad2",  "dias_afiliacion"),
listclass=c("TI_G_enf_totales1", "TI_enf_totales2", "genero_F",
"TI_tipo2", "VIH", "dialisis", "menos1", "oncologia_adultos",
"reumatologia_colagen", "zona_2", "TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=30,
nodesize=30,replace=TRUE,ntree=1000,sampsize=2000,mtry=14)

medias15$modelo="bag6"

#MINER
medias16<-cruzadarf(data=data,
vardep="costo_medio",listconti=c("edad2",  "edad"),
listclass=c("oncologia_adultos",
   "TI_enf_totales2","reumatologia_colageno","dialisis",
"TI_G_enf_totales1"),
grupos=4,sinicio=1234,repe=20,
nodesize=30,replace=TRUE,ntree=1500,sampsize =3000,mtry=7)

medias16$modelo="bag7"

#IMPORTANCIA
medias17<-cruzadarf(data=data,
vardep="costo_medio",listconti=c("edad_M", "edad","edad2"),
listclass=c("TI_G_enf_totales1", "TI_enf_totales1","TI_enf_totales2",
"VIH", "dialisis", "oncologia_adultos"),
grupos=4,sinicio=1234,repe=30,
nodesize=30,replace=TRUE,ntree=1200,sampsize=3000,mtry=9)

medias17$modelo="bag8"

#ALEATORIA
medias18<-cruzadarf(data=data,
vardep="costo medio",listconti=c("edad","edad2",  "dias afiliacion"),
listclass=c("TI_G_enf_totales1", "TI_enf_totales2", "genero_F",
"TI_tipo2", "VIH", "dialisis", "menos1", "oncologia_adultos",
"reumatologia_colagen", "zona_2", "TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=200,
nodesize=30,replace=TRUE,ntree=1000,sampsize=3000,mtry=14)

medias18$modelo="bag9"
```

```
#IMPORTANCIA
medias19<-cruzadarf(data=data,
vardep="costo medio",listconti=c("edad M", "edad","edad2"),
listclass=c("TI_G_enf_totales1", "TI_enf_totales1","TI_enf_totales2",
"VIH", "dialisis", "oncologia_adultos"),
grupos=4,sinicio=1234,repe=30,
nodesize=20,replace=TRUE,ntree=1200,sampsize=2000,mtry=9)

medias19$modelo="bag10"

union1<-rbind(medias10,medias11,medias12, medias13, medias14,
medias15, medias16, medias17, medias18, medias19)

uni<-union1
uni$modelo <- with(uni, reorder(modelo, error, mean))

par(cex.axis=1.2)
boxplot(data=uni,error~modelo, col="gray", main="Error", xlab="",
ylab="")

#RANDOM FOREST
#MINER
medias20<-cruzadarf(data=data,
vardep="costo_medio",listconti=c("edad2",  "edad"),
listclass=c("oncologia_adultos",
   "TI_enf_totales2","reumatologia_colageno","dialisis",
"TI_G_enf_totales1"),
grupos=4,sinicio=1234,repe=200,
nodesize=30,replace=TRUE,ntree=1500,sampsize =2000,mtry=6)

medias20$modelo="RF1"

medias21<-cruzadarf(data=data,
vardep="costo_medio",listconti=c("edad2",  "edad"),
listclass=c("oncologia_adultos",
   "TI_enf_totales2","reumatologia_colageno","dialisis",
"TI_G_enf_totales1"),
grupos=4,sinicio=1234,repe=200,
nodesize=30,replace=TRUE,ntree=1500,sampsize =2000,mtry=5)

medias21$modelo="RF2"

medias22<-cruzadarf(data=data,
vardep="costo_medio",listconti=c("edad2",  "edad"),
listclass=c("oncologia_adultos",
   "TI_enf_totales2","reumatologia_colageno","dialisis",
"TI_G_enf_totales1"),
grupos=4,sinicio=1234,repe=200,
nodesize=30,replace=TRUE,ntree=1500,sampsize =2000,mtry=4)

medias22$modelo="RF3"

#IMPORTANCIA
medias23<-cruzadarf(data=data,
vardep="costo_medio",listconti=c("edad_M", "edad","edad2"),
listclass=c("TI_G_enf_totales1", "TI_enf_totales1","TI_enf_totales2",
"VIH", "dialisis", "oncologia_adultos"),
grupos=4,sinicio=1234,repe=200,
nodesize=30,replace=TRUE,ntree=1200,sampsize=2000,mtry=8)
```

```
medias23$modelo="RF4"

medias24<-cruzadarf(data=data,
vardep="costo medio",listconti=c("edad M", "edad","edad2"),
listclass=c("TI_G_enf_totales1", "TI_enf_totales1","TI_enf_totales2",
"VIH", "dialisis", "oncologia_adultos"),
grupos=4,sinicio=1234,repe=200,
nodesize=30,replace=TRUE,ntree=1200,sampsize=2000,mtry=6)

medias24$modelo="RF5"

medias25<-cruzadarf(data=data,
vardep="costo_medio",listconti=c("edad_M", "edad","edad2"),
listclass=c("TI_G_enf_totales1", "TI_enf_totales1","TI_enf_totales2",
"VIH", "dialisis", "oncologia_adultos"),
grupos=4,sinicio=1234,repe=200,
nodesize=30,replace=TRUE,ntree=1200,sampsize=2000,mtry=4)

medias25$modelo="RF6"

#ALEATORIA
medias26<-cruzadarf(data=data,
vardep="costo_medio",listconti=c("edad","edad2", "dias_afiliacion"),
listclass=c("TI_G_enf_totales1", "TI_enf_totales2", "genero_F",
"TI_tipo2", "VIH", "dialisis", "menos1", "oncologia_adultos",
"reumatologia_colagen", "zona_2", "TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=200,
nodesize=30,replace=TRUE,ntree=1000,sampsize=2000,mtry=12)

medias26$modelo="RF7"

medias27<-cruzadarf(data=data,
vardep="costo_medio",listconti=c("edad","edad2", "dias_afiliacion"),
listclass=c("TI_G_enf_totales1", "TI_enf_totales2", "genero_F",
"TI_tipo2", "VIH", "dialisis", "menos1", "oncologia_adultos",
"reumatologia_colagen", "zona_2", "TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=200,
nodesize=30,replace=TRUE,ntree=1000,sampsize=2000,mtry=10)

medias27$modelo="RF8"

medias28<-cruzadarf(data=data,
vardep="costo_medio",listconti=c("edad","edad2", "dias_afiliacion"),
listclass=c("TI_G_enf_totales1", "TI_enf_totales2", "genero_F",
"TI_tipo2", "VIH", "dialisis", "menos1", "oncologia_adultos",
"reumatologia_colagen", "zona_2", "TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=200,
nodesize=30,replace=TRUE,ntree=1000,sampsize=2000,mtry=8)

medias28$modelo="RF9"

medias29<-cruzadarf(data=data,
vardep="costo_medio",listconti=c("edad","edad2", "dias_afiliacion"),
listclass=c("TI G enf totales1", "TI enf totales2", "genero F",
"TI_tipo2", "VIH", "dialisis", "menos1", "oncologia_adultos",
"reumatologia_colagen", "zona_2", "TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=200,
nodesize=30,replace=TRUE,ntree=1000,sampsize=2000,mtry=6)

medias29$modelo="RF10"
```

```
medias30<-cruzadarf(data=data,
vardep="costo_medio",listconti=c("edad","edad2",  "dias_afiliacion"),
listclass=c("TI_G_enf_totales1",  "TI_enf_totales2", "genero_F",
"TI tipo2", "VIH",  "dialisis", "menos1", "oncologia_adultos",
"reumatologia_colagen", "zona_2", "TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=200,
nodesize=30,replace=TRUE,ntree=1000,sampsize=2000,mtry=4)

medias30$modelo="RF11"

#rf
union1<-rbind(medias20,medias21,medias22,medias23,medias24,medias25,
medias26,  medias27,  medias28,  medias29,  medias30)

uni<-union1
uni$modelo <-with(uni,reorder(modelo,error,mean))

par(cex.axis=0.8)
boxplot(data=uni,error~modelo,col="gray")
```

Incremento gradiente

```
library(caret)
library(dummies)
library(dplyr)
library(pROC)
source("cruzada gbm continua.R")

set.seed(12345)

gbmgrid<-expand.grid(shrinkage=c(0.1,0.05,0.01,0.001),
n.minobsinnode=c(10,20,30),
n.trees=c(1000,2000,3000),
interaction.depth=c(2))

control<-trainControl(method = "cv",number=4,savePredictions = "all")

#MINER
gbm<- train(costo_medio~edad2+ edad+  oncologia_adultos+
   TI_enf_totales2+ reumatologia_colageno+  dialisis+ TI_G_enf_totales1,
data=databis,
method="gbm",trControl=control,tuneGrid=gbmgrid,
distribution="gaussian", bag.fraction=1,verbose=FALSE)

gbm

plot(gbm)
#IMPORTANCIA
gbm2<- train(costo_medio~edad_M+ edad+  edad2+ TI_G_enf_totales1+
   TI_enf_totales1+ TI_enf_totales2+ VIH+ dialisis+ oncologia_adultos,
data=databis,
method="gbm",trControl=control,tuneGrid=gbmgrid,
distribution="gaussian", bag.fraction=1,verbose=FALSE)

gbm2

plot(gbm2)

#ALEATORIA
gbm3<- train(costo_medio~edad+ edad2+ dias_afiliacion+
   TI_G_enf_totales1+ TI_enf_totales2+ genero_F+ TI_tipo2+ VIH+ dialisis+
```

```
  menos1+oncologia_adultos+ reumatologia_colageno+  zona_2+
  TI_OPT_edad4,
data=databis,
method="gbm",trControl=control,tuneGrid=gbmgrid,
distribution="gaussian", bag.fraction=1,verbose=FALSE)

gbm3

plot(gbm3)

# ESTUDIO DE EARLY STOPPING
# Probamos a fijar algunos parÃ¡metros para ver como evoluciona
# en funciÃ³n de las iteraciones

#MINER
gbmgrid<-expand.grid(shrinkage=c(0.01),
n.minobsinnode=c(30),
n.trees=c(500,1000,1200,2000,5000,7000,8000,20000),
interaction.depth=c(2))

control<-trainControl(method = "cv",number=4,savePredictions = "all")

gbm4<- train(costo_medio~edad2+  edad+  oncologia_adultos+
  TI_enf_totales2+ reumatologia_colageno+  dialisis+ TI_G_enf_totales1,
data=databis,
method="gbm",trControl=control,tuneGrid=gbmgrid,
distribution="gaussian", bag.fraction=1,verbose=FALSE)

gbm4

plot(gbm4)


#IMPORTANCIA

gbmgrid<-expand.grid(shrinkage=c(0.001),
n.minobsinnode=c(10),
n.trees=c(500,1000,1200,2000,5000,7000,8000,20000),
interaction.depth=c(2))

control<-trainControl(method = "cv",number=4,savePredictions = "all")

gbm5<- train(costo_medio~edad_M+ edad+  edad2+ TI_G_enf_totales1+
  TI_enf_totales1+ TI_enf_totales2+ VIH+ dialisis+ oncologia_adultos,
data=databis,
method="gbm",trControl=control,tuneGrid=gbmgrid,
distribution="gaussian", bag.fraction=1,verbose=FALSE)

gbm5

plot(gbm5)

#ALEATORIA
gbmgrid<-expand.grid(shrinkage=c(0.01),
n.minobsinnode=c(20),
n.trees=c(500,1000,1200,2000,5000,7000,8000,20000),
interaction.depth=c(2))

control<-trainControl(method = "cv",number=4,savePredictions = "all")
```

```
gbm6<- train(costo_medio~edad+ edad2+ dias_afiliacion+
   TI_G_enf_totales1+ TI_enf_totales2+ genero_F+ TI_tipo2+ VIH+ dialisis+
   menos1+oncologia_adultos+ reumatologia_colageno+  zona_2+
   TI_OPT edad4,
data=databis,
method="gbm",trControl=control,tuneGrid=gbmgrid,
distribution="gaussian", bag.fraction=1,verbose=FALSE)
gbm6
plot(gbm6)

#MINER
medias31<-cruzadagbm(data=data,
vardep="costo_medio",listconti=c("edad2",  "edad"),
listclass=c("oncologia_adultos",
   "TI_enf_totales2","reumatologia_colageno","dialisis",
"TI_G_enf_totales1"),
grupos=4,sinicio=1234,repe=200,
n.minobsinnode=30,shrinkage=0.01,n.trees=1000,interaction.depth=2)

medias31$modelo="gbm"

#IMPORTANCIA
medias32<-cruzadagbm(data=data,
vardep="costo_medio",listconti=c("edad_M", "edad","edad2"),
listclass=c("TI_G_enf_totales1",  "TI_enf_totales1","TI_enf_totales2",
"VIH",  "dialisis", "oncologia_adultos"),
grupos=4,sinicio=1234,repe=200,
n.minobsinnode=10,shrinkage=0.001,n.trees=1200,interaction.depth=2)

medias32$modelo="gbm2"

#ALEATORIA
medias33<-cruzadagbm(data=data,
vardep="costo_medio",listconti=c("edad","edad2",  "dias_afiliacion"),
listclass=c("TI_G_enf_totales1",  "TI_enf_totales2",
"genero_F","TI_tipo2",  "VIH",  "dialisis", "menos1",
   "oncologia_adultos",
"reumatologia_colageno",   "zona_2", "TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=200,
n.minobsinnode=20,shrinkage=0.01,n.trees=1000,interaction.depth=2)

medias33$modelo="gbm3"


union1<-rbind(medias31,medias32,medias33)
uni<-union1
uni$modelo <- with(uni, reorder(modelo, error, mean))


par(cex.axis=1.2)
boxplot(data=uni,error~modelo, col="gray", main="Error", xlab="",
ylab="")
```

Xgboost

```
# VARIABLE CONTINUA
library(caret)
library(dplyr)
library(xgboost)
setwd("C:/")
source ("cruzada xgboost continua.R")
```

```
set.seed(12345)

xgbmgrid<-expand.grid(
min child weight=c(10,20,30),
eta=c(0.01,0.015,0.025,0.05,0.1),
nrounds=c(1000,2000,5000),
max_depth=5,gamma=0,colsample_bytree=1,subsample=1)

# xgbmgrid<-expand.grid(
#   min_child_weight=5,
#   eta=0.005,
#   nrounds=500,
#   max_depth=c(5,7,9,12),gamma=c(0,0.3, 0.5,
0.7,1),colsample_bytree=c(0.8,1),subsample=c(0.8,1))

control<-trainControl(method = "cv",number=4,savePredictions = "all",
classProbs=TRUE)

#MINER
xgbm<- train(costo_medio~edad2+  edad+  oncologia_adultos+
  TI_enf_totales2+ reumatologia_colageno+  dialisis+ TI_G_enf_totales1,
data=databis,
method="xgbTree",trControl=control,
tuneGrid=xgbmgrid,verbose=FALSE)
xgbm
plot(xgbm)

#IMPORTANCIA
xgbm2<- train(costo_medio~edad_M+edad+  edad2+ TI_G_enf_totales1+
  TI_enf_totales1+ TI_enf_totales2+ VIH+ dialisis+ oncologia_adultos,
data=databis,
method="xgbTree",trControl=control,
tuneGrid=xgbmgrid,verbose=FALSE)
xgbm2
plot(xgbm2)

#ALEATORIA
xgbm3<- train(costo_medio~edad+  edad2+ dias_afiliacion+
  TI_G_enf_totales1+ TI_enf_totales2+ genero_F+ TI_tipo2+ VIH+ dialisis+
  menos1+oncologia_adultos+ reumatologia_colageno+  zona_2+
  TI_OPT_edad4,
data=databis,
method="xgbTree",trControl=control,
tuneGrid=xgbmgrid,verbose=FALSE)
xgbm3
plot(xgbm3)

# ESTUDIO DE EARLY STOPPING
# Probamos a fijar algunos parÃ¡metros para ver como evoluciona
# en funciÃ³n de las iteraciones

#MINER
xgbmgrid<-expand.grid(eta=c(0.05),
min child weight=c(30),
nrounds=c(300,500,1000,1500,2000),
max_depth=5,gamma=1,colsample_bytree=1,subsample=1)

set.seed(12345)
control<-trainControl(method = "cv",number=4,savePredictions = "all")
xgbm4<- train(costo_medio~edad2+ edad+  oncologia_adultos+
  TI_enf_totales2+ reumatologia_colageno+  dialisis+ TI_G_enf_totales1,
```

```
data=databis,
method="xgbTree",trControl=control,
tuneGrid=xgbmgrid,verbose=FALSE)
xgbm4
plot(xgbm4)

#IMPORTANCIA
xgbmgrid<-expand.grid(eta=c(0.01),
min_child_weight=c(30),
nrounds=c(300,500,1000,1500,2000),
max_depth=5,gamma=1,colsample_bytree=1,subsample=1)

set.seed(12345)
control<-trainControl(method = "cv",number=4,savePredictions = "all")

xgbm5<- train(costo_medio~edad_M+edad+  edad2+ TI_G_enf_totales1+
  TI_enf_totales1+ TI_enf_totales2+ VIH+ dialisis+ oncologia_adultos,
data=databis,
method="xgbTree",trControl=control,
tuneGrid=xgbmgrid,verbose=FALSE)
xgbm5
plot(xgbm5)

#ALEATORIA
xgbmgrid<-expand.grid(eta=c(0.01),
min_child_weight=c(30),
nrounds=c(300,500,1000,1500,2000),
max_depth=5,gamma=1,colsample_bytree=1,subsample=1)

set.seed(12345)
control<-trainControl(method = "cv",number=4,savePredictions = "all")
xgbm6<- train(costo_medio~edad+  edad2+ dias_afiliacion+
  TI_G_enf_totales1+ TI_enf_totales2+ genero_F+ TI_tipo2+ VIH+ dialisis+
  menos1+oncologia_adultos+ reumatologia_colageno+  zona_2+
  TI_OPT_edad4,
data=databis,
method="xgbTree",trControl=control,
tuneGrid=xgbmgrid,verbose=FALSE)
xgbm6
plot(xgbm6)

# IMPORTANCIA DE VARIABLES

varImp(xgbm)
plot(varImp(xgbm))

# UTILIZACION DE LOS PARAMETROS  DE REGULARIZACION
#VALIDACION CRUZADA
#MINER
medias40<-cruzadaxgbm(data=data,
vardep="costo_medio",listconti=c("edad2",  "edad"),
listclass=c("oncologia_adultos",
  "TI_enf_totales2","reumatologia_colageno","dialisis",
"TI_G_enf_totales1"),
grupos=4,sinicio=1234,repe=200,
min_child_weight=30,eta=0.05,nrounds=300,max_depth=5,
gamma=1,colsample_bytree=1,subsample=1, lambda=0 )

medias40$modelo="xgbm"

#IMPORTANCIA
```

```
medias41<-cruzadaxgbm(data=data,
vardep="costo_medio",listconti=c("edad_M", "edad","edad2"),
listclass=c("TI_G_enf_totales1", "TI_enf_totales1","TI_enf_totales2",
"VIH", "dialisis", "oncologia adultos"),
grupos=4,sinicio=1234,repe=200,
min_child_weight=30,eta=0.01,nrounds=300,max_depth=5,
gamma=1,colsample_bytree=1,subsample=1, lambda=0 )

medias41$modelo="xgbm2"

#ALEATORIA
medias42<-cruzadaxgbm(data=data,
vardep="costo_medio",listconti=c("edad","edad2",  "dias_afiliacion"),
listclass=c("TI_G_enf_totales1", "TI_enf_totales2",
"genero_F","TI_tipo2",  "VIH", "dialisis", "menos1",
   "oncologia_adultos",
"reumatologia_colageno",  "zona_2","TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=200,
min_child_weight=30,eta=0.01,nrounds=300,max_depth=5,
gamma=1,colsample_bytree=1,subsample=1, lambda=0 )

medias42$modelo="xgbm3"

#IMPORTANCIA cambiando numero minimo de observaciones
medias43<-cruzadaxgbm(data=data,
vardep="costo_medio",listconti=c("edad_M", "edad","edad2"),
listclass=c("TI_G_enf_totales1", "TI_enf_totales1","TI_enf_totales2",
"VIH", "dialisis", "oncologia_adultos"),
grupos=4,sinicio=1234,repe=200,
min_child_weight=20,eta=0.01,nrounds=300,max_depth=5,
gamma=1,colsample_bytree=1,subsample=1, lambda=0 )

medias43$modelo="xgbm4"

#IMPORTANCIA sorteo de variables
medias44<-cruzadaxgbm(data=data,
vardep="costo_medio",listconti=c("edad_M", "edad","edad2"),
listclass=c("TI_G_enf_totales1", "TI_enf_totales1","TI_enf_totales2",
"VIH", "dialisis", "oncologia_adultos"),
grupos=4,sinicio=1234,repe=200,
min_child_weight=30,eta=0.01,nrounds=300,max_depth=5,
gamma=1,colsample_bytree=0.8,subsample=1, lambda=0 )

medias44$modelo="xgbm5"

#IMPORTANCIA cambiando sorteo de variables y lambda
medias45<-cruzadaxgbm(data=data,
vardep="costo_medio",listconti=c("edad_M", "edad","edad2"),
listclass=c("TI_G_enf_totales1", "TI_enf_totales1","TI_enf_totales2",
"VIH", "dialisis", "oncologia_adultos"),
grupos=4,sinicio=1234,repe=200,
min_child_weight=30,eta=0.01,nrounds=300,max_depth=5,
gamma=1,colsample_bytree=0.8,subsample=1, lambda=10 )

medias45$modelo="xgbm6"

#IMPORTANCIA cambiando sorteo de variables y obser
medias46<-cruzadaxgbm(data=data,
vardep="costo_medio",listconti=c("edad_M", "edad","edad2"),
listclass=c("TI_G_enf_totales1", "TI_enf_totales1","TI_enf_totales2",
"VIH", "dialisis", "oncologia_adultos"),
```

```
grupos=4,sinicio=1234,repe=200,
min_child_weight=30,eta=0.01,nrounds=300,max_depth=5,
gamma=1,colsample_bytree=0.8,subsample=0.8, lambda=0 )

medias46$modelo="xgbm7"


#IMPORTANCIA cambiando sorteo de variables y obser PROFUNDIDAD 10
medias47<-cruzadaxgbm(data=data,
vardep="costo_medio",listconti=c("edad_M", "edad","edad2"),
listclass=c("TI_G_enf_totales1", "TI_enf_totales1","TI_enf_totales2",
"VIH", "dialisis", "oncologia_adultos"),
grupos=4,sinicio=1234,repe=200,
min_child_weight=30,eta=0.01,nrounds=300,max_depth=10,
gamma=1,colsample_bytree=0.8,subsample=0.8, lambda=0 )

medias47$modelo="xgbm8"

union1<-rbind(medias40,medias41,medias42,medias43, medias44, medias45,
medias46,medias47)

uni<-union1
uni$modelo <- with(uni, reorder(modelo, error, mean))

par(cex.axis=1.2)
boxplot(data=uni,error~modelo, col="gray", main="Error", xlab="",
ylab="")
```

Máquinas de soporte vectorial

```
library(caret)
library(dplyr)
library(dummies)
setwd("C:/")
source("cruzada SVM continua lineal.R")
source("cruzada SVM continua polinomial.R")
source("cruzada SVM continua RBF.R")

# *************************
# TUNEADO SVM CONTINUA
# *************************
#  SVM LINEAL: SOLO PARAMETRO C

SVMgrid<-expand.grid(C=c(0.01,0.05,0.1,0.2,0.5,1,2,5,10))

control<-trainControl(method = "cv",number=4,
savePredictions = "all")

#MINER
SVM<- train(data=databis,
costo_medio~edad2+ edad+  oncologia_adultos+ TI_enf_totales2+
  reumatologia_colageno+  dialisis+ TI_G_enf_totales1,
method="svmLinear",trControl=control,
tuneGrid=SVMgrid,verbose=FALSE)
SVM$results
SVM

#IMPORTANCIA
SVM2<- train(data=databis,
```

```
costo_medio~edad_M+  edad+  edad2+ TI_G_enf_totales1+ TI_enf_totales1+
  TI_enf_totales2+ VIH+ dialisis+ oncologia_adultos,
method="svmLinear",trControl=control,
tuneGrid=SVMgrid,verbose=FALSE)
SVM2$results
SVM2

#ALEATORIA
SVM3<- train(data=databis,
costo_medio~edad+  edad2+ dias_afiliacion+ TI_G_enf_totales1+
  TI_enf_totales2+ genero_F+ TI_tipo2+ VIH+ dialisis+ menos1+
  oncologia_adultos+ reumatologia_colageno+  zona_2+TI_OPT_edad4,
method="svmLinear",trControl=control,
tuneGrid=SVMgrid,verbose=FALSE)
SVM3$results
SVM3

#MINER
medias50<-cruzadaSVM(data=data,
vardep="costo_medio",listconti=c("edad2",  "edad"),
listclass=c("oncologia_adultos",
  "TI_enf_totales2","reumatologia_colageno","dialisis",
"TI_G_enf_totales1"),
grupos=4,sinicio=1234,repe=200,C=10)

medias50$modelo="SVML1"


#IMPORTANCIA
medias51<-cruzadaSVM(data=data,
vardep="costo_medio",listconti=c("edad_M", "edad","edad2"),
listclass=c("TI_G_enf_totales1",  "TI_enf_totales1","TI_enf_totales2",
"VIH",  "dialisis",  "oncologia_adultos"),
grupos=4,sinicio=1234,repe=200,C=0.01)

medias51$modelo="SVML2"

#ALEATORIA
medias52<-cruzadaSVM(data=data,
vardep="costo_medio",listconti=c("edad","edad2",  "dias_afiliacion"),
listclass=c("TI_G_enf_totales1",  "TI_enf_totales2",
"genero_F","TI_tipo2",  "VIH",  "dialisis",  "menos1",
  "oncologia_adultos",
"reumatologia_colageno",  "zona_2", "TI_OPT_edad4"),
grupos=4,sinicio=1234,repe=200,C=5)

medias52$modelo="SVML3"

union1<-rbind(medias50,medias51,medias52)

uni<-union1
uni$modelo <- with(uni, reorder(modelo, error, mean))
par(cex.axis=1.2)
boxplot(data=uni,error~modelo, col="gray", main="Error", xlab="",
ylab="")


#  SVM Polinomial: PARAMETROS C, degree, scale

# SVMgrid<-expand.grid(C=c(0.001,0.01,0.02, 0.03,
0.04,0.05,0.1,1,5,10,20,40),
```

```
#                       degree=c(2,3),scale=c(0.1,0.5,1,2,5))
# control<-trainControl(method = "cv",
#  number=4,savePredictions = "all")
#
#
# SVM<- train(data=databis,
#             costo_medio~dialisis+TI_G_enf_totales1+TI_enf_totales1
#
+TI_enf_totales2+VIH+reumatologia_colageno+edad+edad2+edad_M,
#             method="svmPoly",trControl=control,
#             tuneGrid=SVMgrid,verbose=FALSE)
#
# SVM
#
# SVM$results
#
# plot(SVM$results$C,SVM$results$RMSE)

#  SVM RBF: PARAMETROS C, sigma

SVMgrid<-expand.grid(C=c(0.01,0.05,0.1,0.2,0.5,1,2,5,10),
sigma=c(0.01,0.05,0.1,0.2,0.5,1,2,5,10,30))

control<-trainControl(method = "cv",
number=4,savePredictions = "all")

#MINER
SVM4<- train(data=databis,
costo_medio~edad2+ edad+  oncologia_adultos+ TI_enf_totales2+
  reumatologia_colageno+  dialisis+ TI_G_enf_totales1,
method="svmRadial",trControl=control,
tuneGrid=SVMgrid,verbose=FALSE)

SVM4

#IMPORTANCIA
SVM5<- train(data=databis,
costo_medio~edad_M+  edad+  edad2+ TI_G_enf_totales1+ TI_enf_totales1+
  TI_enf_totales2+ VIH+ dialisis+ oncologia_adultos,
method="svmRadial",trControl=control,
tuneGrid=SVMgrid,verbose=FALSE)

SVM5

#IMPORTANCIA
SVM6<- train(data=databis,
costo_medio~edad+  edad2+ dias_afiliacion+ TI_G_enf_totales1+
  TI_enf_totales2+ genero_F+ TI_tipo2+ VIH+ dialisis+ menos1+
  oncologia_adultos+ reumatologia_colageno+  zona_2+TI_OPT_edad4,
method="svmRadial",trControl=control,
tuneGrid=SVMgrid,verbose=FALSE)

SVM6

data<-databis


#MINER
medias53<-cruzadaSVMRBF(data=data,
vardep="costo_medio",listconti=c("edad2",  "edad"),
```

```
listclass=c("oncologia_adultos",
   "TI_enf_totales2","reumatologia_colageno","dialisis",
"TI_G_enf_totales1"),
grupos=4,sinicio=1234,repe=200,C=10,sigma=0.2)

medias53$modelo="SVMRBF"

#IMPORTANCIA
medias54<-cruzadaSVMRBF(data=data,
vardep="costo_medio",listconti=c("edad_M", "edad","edad2"),
listclass=c("TI_G_enf_totales1", "TI_enf_totales1","TI_enf_totales2",
"VIH", "dialisis", "oncologia_adultos"),
grupos=4,sinicio=1235,repe=200,C=10,sigma=0.1)

medias54$modelo="SVMRBF2"

#ALEATORIA
medias55<-cruzadaSVMRBF(data=data,
vardep="costo_medio",listconti=c("edad","edad2", "dias_afiliacion"),
listclass=c("TI_G_enf_totales1", "TI_enf_totales2",
"genero_F","TI_tipo2", "VIH", "dialisis", "menos1",
   "oncologia_adultos",
"reumatologia_colageno", "zona_2","TI_OPT_edad4"),
grupos=4,sinicio=12345,repe=200,C=10,sigma=0.01)

medias55$modelo="SVMRBF3"

union1<-rbind(medias53,medias54, medias55)

uni<-union1
uni$modelo <- with(uni, reorder(modelo, error, mean))


par(cex.axis=1.2)
boxplot(data=uni,error~modelo, col="gray", main="Error", xlab="",
ylab="")

#EVALUACION
union1<-rbind(medias2,medias5, medias14, medias29, medias33, medias46,
medias52, medias53)

uni<-union1
uni$modelo <- with(uni, reorder(modelo, error, mean))


par(cex.axis=1.2)
boxplot(data=uni,error~modelo, col="gray", main="Error", xlab="",
ylab="")
```

Ensamblado

```
# (variable dependiente continua)
library(dplyr)
library(reshape)
library(MASS)
setwd("C:/")
source("cruzadas ensamblado continuas TFM source.R")

# Por hacer una prueba rapida, comparo regresion con rf en los 3 sets
de variables.
# Lo hago con el esquema de ensamblado, pero todavÃ-a sin ensamblar
```

```
# En cada modelo pongo las variables


archivo<-databis

vardep<-"costo_medio"
listclass<-c("TI_G_enf_totales1","TI_enf_totales1","TI_enf_totales2",
"VIH", "dialisis", "oncologia_adultos")

listconti<-c("edad_M",  "edad","edad2")
grupos<-4
sinicio<-1234
repe<-50

# APLICACION CRUZADAS PARA ENSAMBLAR

medias65<-cruzadalin(data=archivo,
vardep=vardep,listconti=c("edad_M", "edad","edad2"),
listclass=c("TI_G_enf_totales1", "TI_enf_totales1","TI_enf_totales2",
"VIH", "dialisis",
   "oncologia_adultos"),grupos=grupos,sinicio=sinicio,repe=repe)

medias65bis<-as.data.frame(medias65[1])
medias65bis$modelo<-"regresion"
predi65<-as.data.frame(medias65[2])
predi65$reg<-predi65$pred

medias66<-cruzadaavnnet(data=archivo,
vardep=vardep,listconti=c("edad2",  "edad"),
listclass=c("oncologia_adultos",
   "TI_enf_totales2","reumatologia_colageno","dialisis",
"TI_G_enf_totales1"),grupos=grupos,sinicio=sinicio,repe=repe,
size=c(9),decay=c(0.1),repeticiones=5,itera=200,trace=FALSE)

medias66bis<-as.data.frame(medias66[1])
medias66bis$modelo<-"avnnet"
predi66<-as.data.frame(medias66[2])
predi66$avnnet<-predi66$pred


medias67<-cruzadarf(data=archivo,
vardep=vardep,listconti=c("edad","edad2",  "dias_afiliacion"),
listclass=c("TI_G_enf_totales1", "TI_enf_totales2",
"genero_F","TI_tipo2",  "VIH", "dialisis", "menos1",
   "oncologia_adultos",
"reumatologia_colageno",  "zona_2",
   "TI_OPT_edad4"),grupos=grupos,sinicio=sinicio,repe=repe,
mtry=6,ntree=1000,nodesize=30,sampsize=2000,replace=TRUE)


medias67bis<-as.data.frame(medias67[1])
medias67bis$modelo<-"rf"
predi67<-as.data.frame(medias67[2])
predi67$rf<-predi67$pred

medias68<-cruzadagbm(data=archivo,
vardep=vardep,listconti=c("edad","edad2",  "dias_afiliacion"),
listclass=c("TI_G_enf_totales1", "TI_enf_totales2",
"genero_F","TI_tipo2",  "VIH", "dialisis", "menos1",
   "oncologia_adultos",
```

```
"reumatologia_colageno",  "zona_2",
   "TI_OPT_edad4"),grupos=grupos,sinicio=sinicio,repe=repe,
n.minobsinnode=20,shrinkage=0.01,n.trees=1000,interaction.depth=2)

medias68bis<-as.data.frame(medias68[1])
medias68bis$modelo<-"gbm"
predi68<-as.data.frame(medias68[2])
predi68$gbm<-predi68$pred

medias69<-cruzadaxgbm(data=archivo,
vardep=vardep,listconti=c("edad_M", "edad","edad2"),
listclass=c("TI_G_enf_totales1", "TI_enf_totales1","TI_enf_totales2",
"VIH", "dialisis",
   "oncologia_adultos"),grupos=grupos,sinicio=sinicio,repe=repe,
min_child_weight=30,eta=0.01,nrounds=300,max_depth=5,
gamma=1,colsample_bytree=0.8,subsample=0.8, lambda =0)


medias69bis<-as.data.frame(medias69[1])
medias69bis$modelo<-"xgbm"
predi69<-as.data.frame(medias69[2])
predi69$xgbm<-predi69$pred


medias70<-cruzadaSVMRBF(data=archivo,
vardep=vardep,listconti=c("edad2",  "edad"),
listclass=c("oncologia_adultos",
   "TI_enf_totales2","reumatologia_colageno","dialisis",
"TI_G_enf_totales1"),grupos=grupos,sinicio=sinicio,repe=repe,
C=10,sigma=0.2)

medias70bis<-as.data.frame(medias70[1])
medias70bis$modelo<-"SVM"
predi70<-as.data.frame(medias70[2])
predi70$svm<-predi70$pred

# medias71<-cruzadaSVM(data=archivo,
#                          vardep=vardep,listconti=c("edad","edad2",
   "dias_afiliacion"),
#                          listclass=c("TI_G_enf_totales1",
   "TI_enf_totales2", "genero_F","TI_tipo2",  "VIH", "dialisis",
   "menos1", "oncologia_adultos",
#                                      "reumatologia_colageno",
   "zona_2", "TI_OPT_edad4"),grupos=grupos,sinicio=sinicio,repe=repe,
#                          C=10)
#
# medias71bis<-as.data.frame(medias71[1])
# medias71bis$modelo<-"SVMLineal"
# predi71<-as.data.frame(medias71[2])
# predi71$svm<-predi71$pred
#

union1<-rbind(medias65bis,medias66bis,
medias67bis,medias68bis,medias69bis,medias70bis)

par(cex.axis=1)
boxplot(data=union1,error~modelo,col="gray")


# CONSTRUCCION DE TODOS LOS ENSAMBLADOS
```

```
# SE UTILIZARAN LOS ARCHIVOS SURGIDOS DE LAS FUNCIONES LLAMADOS
predi1,...

unipredi<-cbind(predi65,predi66,predi67,predi68, predi69, predi70)

# Esto es para eliminar columnas duplicadas
unipredi<- unipredi[, !duplicated(colnames(unipredi))]

# Construccion de ensamblados, cambiar al gusto

unipredi$predi81<-(unipredi$reg+unipredi$avnnet)/2
unipredi$predi82<-(unipredi$reg+unipredi$rf)/2
unipredi$predi83<-(unipredi$reg+unipredi$gbm)/2
unipredi$predi84<-(unipredi$reg+unipredi$xgbm)/2
unipredi$predi85<-(unipredi$reg+unipredi$svm)/2
unipredi$predi86<-(unipredi$avnnet+unipredi$rf)/2
unipredi$predi87<-(unipredi$avnnet+unipredi$gbm)/2
unipredi$predi88<-(unipredi$avnnet+unipredi$xgbm)/2
unipredi$predi89<-(unipredi$avnnet+unipredi$svm)/2
unipredi$predi90<-(unipredi$rf+unipredi$gbm)/2
unipredi$predi91<-(unipredi$rf+unipredi$xgbm)/2
unipredi$predi92<-(unipredi$rf+unipredi$svm)/2
unipredi$predi93<-(unipredi$gbm+unipredi$xgbm)/2
unipredi$predi94<-(unipredi$gbm+unipredi$svm)/2
unipredi$predi95<-(unipredi$xgbm+unipredi$svm)/2

unipredi$predi96<-(unipredi$reg+unipredi$avnnet+unipredi$rf)/3
unipredi$predi97<-(unipredi$reg+unipredi$avnnet+unipredi$gbm)/3
unipredi$predi98<-(unipredi$reg+unipredi$avnnet+unipredi$xgbm)/3
unipredi$predi99<-(unipredi$reg+unipredi$avnnet+unipredi$svm)/3

unipredi$predi100<-(unipredi$reg+unipredi$rf+unipredi$gbm)/3
unipredi$predi101<-(unipredi$reg+unipredi$rf+unipredi$xgbm)/3
unipredi$predi102<-(unipredi$reg+unipredi$rf+unipredi$svm)/3
unipredi$predi103<-(unipredi$rf+unipredi$avnnet+unipredi$gbm)/3
unipredi$predi104<-(unipredi$rf+unipredi$gbm+unipredi$xgbm)/3
unipredi$predi105<-(unipredi$rf+unipredi$gbm+unipredi$xgbm)/3
unipredi$predi106<-(unipredi$svm+unipredi$gbm+unipredi$xgbm)/3
unipredi$predi107<-(unipredi$reg+unipredi$gbm+unipredi$xgbm)/3
unipredi$predi108<-(unipredi$reg+unipredi$gbm+unipredi$svm)/3
unipredi$predi109<-(unipredi$reg+unipredi$xgbm+unipredi$svm)/3
unipredi$predi110<-
(unipredi$reg+unipredi$rf+unipredi$gbm+unipredi$avnnet)/4
unipredi$predi111<-
(unipredi$reg+unipredi$rf+unipredi$gbm+unipredi$xgbm)/4
unipredi$predi112<-
(unipredi$reg+unipredi$svm+unipredi$gbm+unipredi$xgbm)/4
unipredi$predi113<-
(unipredi$reg+unipredi$avnnet+unipredi$gbm+unipredi$xgbm)/4
unipredi$predi114<-
(unipredi$reg+unipredi$avnnet+unipredi$svm+unipredi$xgbm)/4
unipredi$predi115<-
(unipredi$reg+unipredi$rf+unipredi$gbm+unipredi$xgbm)/4

dput(names(unipredi))


listado<-c("reg", "avnnet", "rf", "gbm", "xgbm", "svm",
"predi81", "predi82", "predi83",

"predi84", "predi85", "predi86", "predi87", "predi88", "predi89",
```

```
"predi90", "predi91", "predi92", "predi93", "predi94", "predi95",

"predi96", "predi97", "predi98", "predi99", "predi100", "predi101",

"predi102", "predi103", "predi104", "predi105", "predi106",
"predi107",

"predi108", "predi109", "predi110", "predi111", "predi112",
"predi113",

"predi114", "predi115")


repeticiones<-nlevels(factor(unipredi$Rep))
unipredi$Rep<-as.factor(unipredi$Rep)
unipredi$Rep<-as.numeric(unipredi$Rep)

# Calculo el MSE para cada repeticion de validaciÃ³n cruzada

medias0<-data.frame(c())

for (prediccion in listado)
{
paso <-unipredi[,c("obs",prediccion,"Rep")]
paso$error<-(paso[,c(prediccion)]-paso$obs)^2
paso<-paso %>%
group_by(Rep) %>%
summarize(error=mean(error))
paso$modelo<-prediccion
medias0<-rbind(medias0,paso)
}
# Finalmente boxplot

par(cex.axis=0.8,las=2)
boxplot(data=medias0,outcex=0.3,error~modelo)

# PRESENTACION TABLA MEDIAS

tablamedias<-medias0 %>%
summarize(error=mean(error))

tablamedias<-tablamedias[order(tablamedias$error),]

# ORDENACIÃ‘N DEL FACTOR MODELO POR LAS MEDIAS EN ERROR
# PARA EL GRÃ FICO

medias0$modelo <- with(medias0,
reorder(modelo,error, mean))
par(cex.axis=0.7,las=2)
boxplot(data=medias0,error~modelo,col="gray")

#GRAFICOS PARA OBSERVAR PREDICCIONES DE DIFERENTES ALGORITMOS

unipredi<-cbind(predi55,predi56,predi57,predi58,predi59,predi60)
# Esto es para eliminar columnas duplicadas
unipredi<- unipredi[, !duplicated(colnames(unipredi))]

# Correlaciones entre predicciones de cada algoritmo individual

#unigraf<-unipredi unipredi$Rep=="Rep001",]
```

```
solos<-c("logi", "avnnet",
"rf","gbm",  "xgbm", "svm")

mat<-unigraf[,solos]
matrizcorr<-cor(mat)
matrizcorr


library(corrplot)
corrplot(matrizcorr, type = "upper", order = "hclust",
tl.col ="black", tl.srt = 45,cl.lim=c(0.7,1),is.corr=FALSE)
```

Resultados

```
library(caret)
library(glmnet)
library(matrix)

costefinal$TI_G_enf_totales1<-factor(costefinal$TI_G_enf_totales1)
costefinal$TI_enf_totales1<-factor(costefinal$TI_enf_totales1)
costefinal$TI_enf_totales2<-factor(costefinal$TI_enf_totales2)
costefinal$VIH<-factor(costefinal$VIH)
costefinal$dialisis<-factor(costefinal$dialisis)
costefinal$oncologia_adultos<-factor(costefinal$oncologia_adultos)

set.seed(2784)
partitionIndex <- createDataPartition(costefinal$costo_medio, p=0.8,
list=FALSE)
data_train <- costefinal[partitionIndex,]
data_test <- costefinal[-partitionIndex,]

modelo2<-lm(costo_medio~edad_M+  edad+  edad2+ TI_G_enf_totales1+
  TI_enf_totales2+ VIH+ dialisis+ oncologia_adultos,
data=data_train[,1:38])
summary(modelo2)
coef(modelo2)

predict(modelo2,costetotal)
setwd("C:/ ")
prediccionTotal<-predict(modelo2,costetotal)
prediccionestotal<-costetotal
prediccionestotal$prediccion<-prediccionTotal
write.table(prediccionestotal,file="prediccionestotal.csv",
col.names=TRUE)

Rsq<-function(modelo,varObj,datos){
testpredicted<-predict(modelo, datos)
5
testReal<-datos[,varObj]
sse <- sum((testpredicted - testReal) ^ 2)
sst <- sum((testReal - mean(testReal)) ^ 2)
1 - sse/sst
}
```

Códigos utilizados en SAS Base

Modelo de dos partes Variable objetivo binaria (Parte 1)

```
libname discoc 'C:\ ';
data Binario;set discoc.costobinario;run;
```

```
proc freq data=binario;run;
proc contents data=binario out=sal;run;quit;
data;set sal;put name @@;run;
options mprint=0;

/* costo_binario

continuas edad_F edad_M dias_afiliacion edad edad2

TI_edad21 TI_edad22 TI_edad23 TI_G_enf_totales1 TI_G_enf_totales0
TI_dias_afil1 TI_dias_afil2 TI_dias_afil3 TI_OPT_edad21 TI_OPT_edad22
TI_OPT_edad23 TI_OPT_edad24
TI_enf_totales1 TI_enf_totales2 TI_enf_totales3 TI_estado_afiliado1
TI_estado_afiliado2 genero_F
M TI_tipo1 TI_tipo2 TI_tipo3 VIH dialisis  mas1 menos1
oncologia_adultos reumatologia_colageno
zona_1 zona_10 zona_11 zona_2 zona_4 zona_5 zona_9
*/

ods output type3=parametros;
proc logistic data=binario namelen=20 descending ;
class ;
model costo_binario=edad_F edad_M dias_afiliacion dias_afil_porc edad
edad2 TI_edad21 TI_edad22
TI_G_enf_totales1 TI_dias_afil1 TI_dias_afil2 TI_OPT_edad21
TI_OPT_edad22 TI_OPT_edad24 TI_enf_totales1
TI_enf_totales2 TI_estado_afiliado1 genero_F TI_tipo1 TI_tipo2 VIH
dialisis mas1 menos1 oncologia_adultos
reumatologia_colageno zona_1 zona_10 zona_11 zona_2 zona_4 zona_5
zona_9 TI_OPT_edad1  TI_OPT_edad2  TI_OPT_edad3  TI_OPT_edad4
/selection=stepwise;
run;quit;
data mode;length length effect $20. modelo $ 20000;retain modelo "
";set parametros end=fin;effect=cat(' ',effect);
if _n_ ne 1 then modelo=catt(modelo,' ',effect);if fin then output;
run;
data ;set mode;put modelo;run;


ods output bestsubsets=modelos;
proc logistic data=binario descending;
model costo_binario=edad_F edad_M dias_afiliacion dias_afil_porc edad
edad2 TI_edad21 TI_edad22
TI_G_enf_totales1 TI_dias_afil1 TI_dias_afil2 TI_OPT_edad21
TI_OPT_edad22 TI_OPT_edad24 TI_enf_totales1
TI_enf_totales2 TI_estado_afiliado1 genero_F TI_tipo1 TI_tipo2 VIH
dialisis menos1 oncologia_adultos
reumatologia_colageno zona_1 zona_10 zona_11 zona_2 zona_4 zona_5
zona_9 TI_OPT_edad1 TI_OPT_edad2 TI_OPT_edad3 TI_OPT_edad4
/selection=score best=1 start=6 stop=10;
run;
data ;set modelos;put variablesinmodel;run;


/*seleccion de variables */

%macro
randomselectlog(data=,listclass=,vardepen=,modelo=,sinicio=,sfinal=,fr
acciontrain=,directorio=);
options nocenter linesize=256;
```

```
proc printto print="&directorio\kk.txt";run;
data;file "&directorio\cosa2.txt" ;run;
%do semilla=&sinicio %to &sfinal;
proc surveyselect data=&data rate=&fracciontrain out=sal1234
seed=&semilla;run;

%if &listclass ne %then %do;
ods output type3=parametros;
proc logistic data=sal1234;
class &listclass;
model &vardepen= &modelo/ selection=stepwise;
run;
data parametros;length effect $20. modelo $ 20000;retain modelo "
";set parametros end=fin;effect=cat(' ',effect);
if _n_ ne 1 then modelo=catt(modelo,' ',effect);if fin then
do;variable=modelo;output;end;
run;
%end;
%else %do;
ods output  Logistic.ParameterEstimates=parametros;
proc logistic data=sal1234;
model &vardepen= &modelo/ selection=stepwise;
run;
%end;
ods graphics off;
ods html close;
data;file "&directorio\cosa2.txt" mod;set parametros;
%if &listclass ne %then %do; put variable @@;%end;
%else %do; if _n_ ne 1 then put variable @@;%end;
run;
%end;
proc printto ;run;
data todos;
infile "&directorio\cosa2.txt";
length efecto $ 400;
input efecto @@;
if efecto ne 'Intercept' then output;
run;
proc freq data=todos;tables efecto /out=sal;run;
proc sort data=sal;by descending count;
proc print data=sal;run;

data todos;
infile "&directorio\cosa2.txt";
length efecto $ 300;
input efecto $ &&;
run;
proc freq data=todos;tables efecto /out=sal;run;
proc sort data=sal;by descending count;
proc print data=sal;run;
data;set sal;put efecto;run;
%mend;


%randomselectlog(data=binario,directorio=discoc,
listclass=TI_edad21 TI_edad22 TI_G_enf_totales1 TI_dias_afil1
TI_dias_afil2 TI_OPT_edad21 TI_OPT_edad22 TI_OPT_edad24
TI_enf_totales1 TI_enf_totales2 TI_estado_afiliado1 genero_F TI_tipo1
TI_tipo2 VIH dialisis menos1 oncologia_adultos reumatologia_colageno
zona_1 zona_10 zona_11 zona_2 zona_4 zona_5 zona_9 TI_OPT_edad1
  TI_OPT_edad2  TI_OPT_edad3  TI_OPT_edad4,
```

```
vardepen=costo_binario,
modelo=edad_F edad_M dias_afiliacion dias_afil_porc edad edad2
TI_edad21 TI_edad22 TI_G_enf_totales1 TI_dias_afil1 TI_dias_afil2
TI OPT edad21 TI OPT edad22 TI OPT edad24 TI enf totales1
TI_enf_totales2 TI_estado_afiliado1 genero_F TI_tipo1 TI_tipo2 VIH
dialisis menos1 oncologia_adultos reumatologia_colageno zona_1 zona_10
zona_11 zona_2 zona_4 zona_5 zona_9 TI_OPT_edad1  TI_OPT_edad2
  TI_OPT_edad3  TI_OPT_edad4,
sinicio=12345,sfinal=12545,fracciontrain=0.8);

options nonotes;
options notes;


/* Regresiones logisticas */

%macro
cruzadalogistica(archivo=,vardepen=,conti=,categor=,ngrupos=,sinicio=,
sfinal=,objetivo=tasafallos);
title ' ';
data final;run;
/* Bucle semillas */
%do semilla=&sinicio %to &sfinal;
data dos;set &archivo;u=ranuni(&semilla);
proc sort data=dos;by u;run;
data dos (drop=nume);
retain grupo 1;
set dos nobs=nume;
if _n_>grupo*nume/&ngrupos then grupo=grupo+1;
run;
data fantasma;run;
%do exclu=1 %to &ngrupos;
data tres;set dos;if grupo ne &exclu then vardep=&vardepen;
proc logistic data=tres noprint;/*<<<<<<******SE PUEDE QUITAR EL
NOPRINT */
%if (&categor ne) %then %do;class &categor;model vardep=&conti
&categor ;%end;
%else %do;model vardep=&conti;%end;
output out=sal p=predi;run;
data sal2;set sal;pro=1-predi;if pro>0.5 then pre11=1; else pre11=0;
if grupo=&exclu then output;run;
proc freq data=sal2;tables pre11*&vardepen/out=sal3;run;
data estadisticos (drop=count percent pre11 &vardepen);
retain vp vn fp fn suma 0;
set sal3 nobs=nume;
suma=suma+count;
if pre11=0 and &vardepen=0 then vn=count;
if pre11=0 and &vardepen=1 then fn=count;
if pre11=1 and &vardepen=0 then fp=count;
if pre11=1 and &vardepen=1 then vp=count;
if _n_=nume then do;
porcenVN=vn/suma;
porcenFN=FN/suma;
porcenVP=VP/suma;
porcenFP=FP/suma;
sensi=vp/(vp+fn);
especif=vn/(vn+fp);
tasafallos=1-(vp+vn)/suma;
tasaciertos=1-tasafallos;
precision=vp/(vp+fp);
F_M=2*Sensi*Precision/(Sensi+Precision);
```

```
output;
end;
run;

data fantasma;set fantasma estadisticos;run;
%end;
proc means data=fantasma sum noprint;var &objetivo;
output out=sumaresi sum=suma mean=media;
run;
data sumaresi;set sumaresi;semilla=&semilla;
data final (keep=suma media semilla);set final sumaresi;if suma=. then
delete;run;
%end;
proc print data=final;run;
%mend;

/* SELECCION MINER */
%cruzadalogistica
(archivo=binario,vardepen=costo_binario,
conti=edad_F edad,
categor=TI_dias_afil1 genero_F TI_tipo2 TI_estado_afiliado1 TI_edad21
TI_G_enf_totales1
TI_OPT_edad2 TI_OPT_edad3 zona_9 zona_2  zona_4 zona_5,
ngrupos=4,sinicio=12345,sfinal=12445);
data final1;set final;modelo=1;

/* SELECCION IMPORTANCIA DE LA VARIABLE */
%cruzadalogistica
(archivo=binario,vardepen=costo_binario,
conti=edad  edad2  edad_M edad_F,
categor=TI_edad21 TI_G_enf_totales1 TI_enf_totales1 TI_enf_totales2,
ngrupos=4,sinicio=12345,sfinal=12445);
data final2;set final;modelo=2;

/* RANDOM SELECT 1*/
%cruzadalogistica
(archivo=binario,vardepen=costo_binario,
conti=dias_afiliacion,
categor=TI_edad21 TI_G_enf_totales1 TI_dias_afil1 TI_dias_afil2
  TI_estado_afiliado1
genero_F TI_tipo2 zona_1 zona_4  zona_5 zona_9 TI_OPT_edad1
TI_OPT_edad4,
ngrupos=4,sinicio=12345,sfinal=12445);
data final3;set final;modelo=3;

/* RANDOM SELECT 2*/
%cruzadalogistica
(archivo=binario,vardepen=costo_binario,
conti=dias_afil_porc,
categor=TI_edad21 TI_edad22  TI_G_enf_totales1 TI_estado_afiliado1
genero_F
TI_tipo2 zona_1  zona_4 zona_5 zona_9 TI_OPT_edad1 TI_OPT_edad4,
ngrupos=4,sinicio=12345,sfinal=12445);
data final4;set final;modelo=4;


/*MEJOR CON 10*/
%cruzadalogistica
(archivo=binario,vardepen=costo_binario,
conti=edad_F  edad_M edad2,
```

```
categor=TI_G_enf_totales1 TI_dias_afil1 TI_tipo2  zona_1 zona_5 zona_9
  TI_OPT_edad2,
ngrupos=4,sinicio=12345,sfinal=12445);
data final5;set final;modelo=5;

data union;set final1 final2 final3 final4 final5;
proc boxplot data=union;plot media*modelo;run;

/*REDES */

/* NUMERO DE NODOS - SET SELECCION MINER

%macro
variar(seminicio=,semifin=,inicionodos=,finalnodos=,increnodos=);
title '';
data union;run;
%do semilla=&seminicio %to &semifin;
%do nodos=&inicionodos %to &finalnodos %by &increnodos;
%neuralbinariabasica(archivo=binario,
listconti=edad_F edad,
listclass=TI_dias_afil1 genero_F TI_tipo2 TI_estado_afiliado1 TI_edad21
TI_G_enf_totales1 TI_OPT_edad2 TI_OPT_edad3 zona_9  zona_2 zona_4
  zona_5,
vardep=costo_binario,nodos=&nodos,corte=50,semilla=&semilla,porcen=0.8
0,algo=levmar);
data estadisticos;set estadisticos;nodos=&nodos;semilla=&semilla;run;
data union;set union estadisticos;run;
%end;
%end;
proc sort data=union;by nodos;run;
proc boxplot data=union;plot (porcenVN porcenFN porcenVP porcenFP
sensi especif tasafallos tasaciertos precision F_M)*nodos;run;
%mend;

%variar(seminicio=12345,semifin=12445,inicionodos=3,finalnodos=11,incr
enodos=2);

/* CON BPROP
%macro
variar(seminicio=,semifin=,inicionodos=,finalnodos=,increnodos=);
title '';
data union;run;
%do semilla=&seminicio %to &semifin;
%do nodos=&inicionodos %to &finalnodos %by &increnodos;
%neuralbinariabasica(archivo=binario,
listconti=edad_F edad2  edad_M,
listclass=TI_G_enf_totales1  TI_edad21 TI_tipo2  zona_5 zona_9
  oncologia_adultos  TI_estado_afiliado1  TI_dias_afil1  F  zona_1
  zona_11zona_4,
vardep=costo_binario,nodos=&nodos,corte=50,semilla=&semilla,porcen=0.8
0,algo=bprop mom=0.2 learn=0.01);
data estadisticos;set estadisticos;nodos=&nodos;semilla=&semilla;run;
data union;set union estadisticos;run;
%end;
%end;
proc sort data=union;by nodos;run;
proc boxplot data=union;plot (porcenVN porcenFN porcenVP porcenFP
sensi especif tasafallos tasaciertos precision F_M)*nodos;run;
%mend;
```

```
%variar(seminicio=12345,semifin=12375,inicionodos=3,finalnodos=7,incre
nodos=1);

*/

/* NUMERO DE NODOS - SET IMPORTANCIA

%macro
variar(seminicio=,semifin=,inicionodos=,finalnodos=,increnodos=);
title '';
data union;run;
%do semilla=&seminicio %to &semifin;
%do nodos=&inicionodos %to &finalnodos %by &increnodos;
%neuralbinariabasica(archivo=binario,
listconti=edadedad2  edad_M edad_F,
listclass=TI_edad21 TI_G_enf_totales1 TI_enf_totales1 TI_enf_totales2,
vardep=costo_binario,nodos=&nodos,corte=50,semilla=&semilla,porcen=0.8
0,algo=levmar);
data estadisticos;set estadisticos;nodos=&nodos;semilla=&semilla;run;
data union;set union estadisticos;run;
%end;
%end;
proc sort data=union;by nodos;run;
proc boxplot data=union;plot (porcenVN porcenFN porcenVP porcenFP
sensi especif tasafallos tasaciertos precision F_M)*nodos;run;
%mend;

%variar(seminicio=12345,semifin=12445,inicionodos=3,finalnodos=18,incr
enodos=2);

/* NUMERO DE NODOS - SET RANDOM SELECT 2

%macro
variar(seminicio=,semifin=,inicionodos=,finalnodos=,increnodos=);
title '';
data union;run;
%do semilla=&seminicio %to &semifin;
%do nodos=&inicionodos %to &finalnodos %by &increnodos;
%neuralbinariabasica(archivo=binario,
listconti=dias_afil_porc  ,
listclass=TI_edad21  TI_edad22 TI_G_enf_totales1  TI_estado_afiliado1
  genero_F TI_tipo2
zona_1  zona_4  zona_5  zona_9  TI_OPT_edad1  TI_OPT_edad4,
vardep=costo_binario,nodos=&nodos,corte=50,semilla=&semilla,porcen=0.8
0,algo=levmar);
data estadisticos;set estadisticos;nodos=&nodos;semilla=&semilla;run;
data union;set union estadisticos;run;
%end;
%end;
proc sort data=union;by nodos;run;
proc boxplot data=union;plot (porcenVN porcenFN porcenVP porcenFP
sensi especif tasafallos tasaciertos precision F_M)*nodos;run;
%mend;

%variar(seminicio=12345,semifin=12395,inicionodos=3,finalnodos=12,incr
enodos=2);


/* NUMERO DE NODOS - SET RANDOM SELECT 1
```

```
%macro
variar(seminicio=,semifin=,inicionodos=,finalnodos=,increnodos=);
title '';
data union;run;
%do semilla=&seminicio %to &semifin;
%do nodos=&inicionodos %to &finalnodos %by &increnodos;
%neuralbinariabasica(archivo=binario,
listconti=dias_afiliacion,
listclass=TI_edad21  TI_G_enf_totales1  TI_dias_afil1 TI_dias_afil2
  TI_estado_afiliado1  genero_F
TI_tipo2  zona_1 zona_4 zona_5 zona_9 TI_OPT_edad1  TI_OPT_edad4,
vardep=costo_binario,nodos=&nodos,corte=50,semilla=&semilla,porcen=0.8
0,algo=levmar);
data estadisticos;set estadisticos;nodos=&nodos;semilla=&semilla;run;
data union;set union estadisticos;run;
%end;
%end;
proc sort data=union;by nodos;run;
proc boxplot data=union;plot (porcenVN porcenFN porcenVP porcenFP
sensi especif tasafallos tasaciertos precision F_M)*nodos;run;
%mend;

%variar(seminicio=12345,semifin=12445,inicionodos=3,finalnodos=11,incr
enodos=2);


/* NUMERO DE NODOS - MEJOR CON 10

%macro
variar(seminicio=,semifin=,inicionodos=,finalnodos=,increnodos=);
title '';
data union;run;
%do semilla=&seminicio %to &semifin;
%do nodos=&inicionodos %to &finalnodos %by &increnodos;
%neuralbinariabasica(archivo=binario,
listconti=edad_F edad_M edad2,
listclass=TI_G_enf_totales1  TI_dias_afil1  TI_tipo2  zona_1 zona_5
  zona_9  TI_OPT_edad2,
vardep=costo_binario,nodos=&nodos,corte=50,semilla=&semilla,porcen=0.8
0,algo=levmar);
data estadisticos;set estadisticos;nodos=&nodos;semilla=&semilla;run;
data union;set union estadisticos;run;
%end;
%end;
proc sort data=union;by nodos;run;
proc boxplot data=union;plot (porcenVN porcenFN porcenVP porcenFP
sensi especif tasafallos tasaciertos precision F_M)*nodos;run;
%mend;

%variar(seminicio=12345,semifin=12445,inicionodos=3,finalnodos=15,incr
enodos=3);


/* EARLY STOPPING */
%macro cruzadabinarianeuralearly(archivo=,vardepen=,
conti=,categor=,ngrupos=,sinicio=,sfinal=,
nodos=,meto=levmar,objetivo=tasafallos,directorio=c:,early=);
data final;run;
proc printto print="directorio\basura.txt"; run;

/* Bucle semillas */
```

```
%do semilla=&sinicio %to &sfinal;
data dos;set &archivo;u=ranuni(&semilla);
proc sort data=dos;by u;run;
data dos (drop=nume);
retain grupo 1;
set dos nobs=nume;
if _n_>grupo*nume/&ngrupos then grupo=grupo+1;
run;
data fantasma;run;
%do exclu=1 %to &ngrupos;

data trestr tresval;
set dos;if grupo ne &exclu then output trestr;else output tresval;
PROC DMDB DATA=trestr dmdbcat=catatres;
target &vardepen;
var &conti;
class &vardepen;
%if &categor ne %then %do;class &categor &vardepen;%end;
run;
proc neural data=trestr dmdbcat=catatres random=789 ;
input &conti;
%if &categor ne %then %do;input &categor /level=nominal;%end;
target &vardepen /level=nominal;
hidden &nodos /act=tanh;/*<<<<<******PARA DATOS LINEALES ACT=LIN
(funci?nde activaci?n lineal)
NORMALMENTE PARA DATOS NO LINEALES MEJOR ACT=TANH */
/* A PARTIR DE AQUI SON ESPECIFICACIONES DE LA RED, SE PUEDEN CAMBIAR
O A?ADIR COMO PAR?METROS */

/*nloptions maxiter=500*/;
netoptions randist=normal ranscale=0.15 random=15459;
train maxiter=&early outest=mlpest technique=&meto;
score data=tresval role=valid out=sal ;
run;
data sal2;set sal;pro=1-%str(p_&vardepen)0;if pro>0.5 then pre11=1;
else pre11=0;run;
proc freq data=sal2;tables pre11*&vardepen/out=sal3;run;

data estadisticos (drop=count percent pre11 &vardepen);
retain vp vn fp fn suma 0;
set sal3 nobs=nume;
suma=suma+count;
if pre11=0 and &vardepen=0 then vn=count;
if pre11=0 and &vardepen=1 then fn=count;
if pre11=1 and &vardepen=0 then fp=count;
if pre11=1 and &vardepen=1 then vp=count;
if _n_=nume then do;
/* porcenVN=vn/suma;
porcenFN=FN/suma;
porcenVP=VP/suma;
porcenFP=FP/suma;
sensi=vp/(vp+fn);
especif=vn/(vn+fp);
precision=vp/(vp+fp);
F_M=2*Sensi*Precision/(Sensi+Precision);
*/
tasafallos=1-(vp+vn)/suma;
tasaciertos=1-tasafallos;
output;
end;
run;
```

```
data fantasma;set fantasma estadisticos;run;
%end;
proc means data=fantasma sum noprint;var &objetivo;
output out=sumaresi sum=suma mean=media;
run;
data sumaresi;set sumaresi;semilla=&semilla;
data final (keep=suma media semilla);set final sumaresi;if suma=. then
delete;run;
%end;
proc printto ;
proc print data=final;run;
%mend;

%macro early1(archivo=,vardepen=,
conti=,categor=,ngrupos=,sinicio=,sfinal=,
nodos=,meto=,objetivo=,directorio=,inicio=,final=,incremento=);
data union;run;
%do early=&inicio %to &final %by &incremento;
%cruzadabinarianeuralearly(archivo=&archivo,vardepen=&vardepen,
conti=&conti,categor=&categor,ngrupos=&ngrupos,sinicio=&sinicio,sfinal
=
&sfinal,nodos=&nodos,meto=&meto,objetivo=&objetivo,directorio=&directo
rio,early=&early);
data final;set final;early=&early;run;
data union;set union final;run;
%end;
title1
h=1.5 j=c c=black "semilla=&sinicio"
h=1 j=c c=green "NODOS OCULTOS: &nodos " " METODO: &meto "  ;
;
symbol1 c=red v=circle i=join pointlabel=("#tasafallos" h=1 c=red
position=bottom  j=c);

proc gplot data=union;plot media*early;run;
%mend;


%macro
neuralbinariabasica(archivo=,listconti=,listclass=,vardep=,nodos=,cort
e=,semilla=,porcen=,algo=levmar);
title '';
data archivobase;set &archivo nobs=nume;ene=int(&porcen*nume);
call symput('ene',left(ene));
run;

proc sort data=archivobase;by &vardep;run;

proc surveyselect data=archivobase out=muestra outall N=&ene
seed=&semilla;
/*si se quiere estratificacion en el muestreo quitar los comentarios
en strata*/
/* strata &vardep /alloc=proportional;*/run;
data train valida;set muestra;if selected=1 then output train;else
output valida;run;

PROC DMDB DATA=train dmdbcat=cataprueba;
target &vardep;
var &listconti;
class &listclass &vardep;
run;
```

```
%if &listclass ne %then %do;
proc neural data=train dmdbcat=cataprueba;
input  &listconti;
input &listclass /level=nominal;
target  &vardep /level=nominal;
hidden &nodos;
prelim 5;
train tech=&algo;
score data=valida out=salpredi outfit=salfit ;
run;
%end;

%else %do;
proc neural data=train dmdbcat=cataprueba;
input  &listconti;
target  &vardep /level=nominal;
hidden &nodos;
prelim 5;
train tech=&algo;
score data=valida out=salpredi outfit=salfit ;
run;
%end;

data salpredi;set salpredi;if p_&vardep.1>&corte/100 then
predi1=1;else predi1=0;run;
proc freq data=salpredi;tables predi1*&vardep/out=sal1;run;

/* Cálculo de estadísticos */

data estadisticos (drop=count percent predi1 &vardep);
retain vp vn fp fn suma 0;
set sal1 nobs=nume;
suma=suma+count;
if predi1=0 and &vardep=0 then vn=count;
if predi1=0 and &vardep=1 then fn=count;
if predi1=1 and &vardep=0 then fp=count;
if predi1=1 and &vardep=1 then vp=count;
if _n_=nume then do;
if vn=. then vn=0;if fn=. then fn=0;if vp=. then vp=0;if fp=. then
fp=0;
porcenVN=vn/suma;
porcenFN=FN/suma;
porcenVP=VP/suma;
porcenFP=FP/suma;
sensi=vp/(vp+fn);
especif=vn/(vn+fp);
tasafallos=1-(vp+vn)/suma;
tasaciertos=1-tasafallos;
precision=vp/(vp+fp);
if vp=0 then precision=0;
if vp=0 then sensi=0;
if vn=0 then especif=0;
F_M=2*Sensi*Precision/(Sensi+Precision);
output;
end;
run;
proc print data=estadisticos;run;

%mend;
```

```
%macro
redneuronalbinaria(archivo=,listclass=,listconti=,vardep=,porcen=,semi
lla=,ocultos=,meto=levmar,acti=);

PROC DMDB DATA=&archivo dmdbcat=catauno;
target &vardep;
var &listconti ;
class &vardep &listclass;
run;

data ooo;set &archivo;run;
data datos;set ooo nobs=nume;tr=int(&porcen*nume);call
symput('tr',left(tr));u=ranuni(&semilla);run;
proc sort data=datos;by u;run;
data datos valida;set datos;if _n_>tr then output valida;else output
datos;run;

proc neural data=datos dmdbcat=catauno validata=valida graph;
input &listconti / id=i;
input &listclass / level=nominal;
target &vardep / level=nominal id=o error=ENT;
hidden &ocultos / id=h act=&acti;
nloptions maxiter=10000;
netoptions randist=normal ranscale=0.1 random=15115;
prelim 0;
train maxiter=10000 outest=mlpest estiter=1 technique=&meto;
score data=datos out=mlpout outfit=mlpfit;
score data=valida out=mlpout2 outfit=mlpfit2 role=valid;
run;

data mlpest2 ;
k=3;
retain iterepocas 0;
set mlpest nobs=nume;
call symput('numeroit',left(nume));
eval=_VOBJERR_;
x3=lag3(eval);
x6=lag6(eval);
if _n_>6 and eval>x3 and eval>x6 then iterepocas=_n_;
run;

data;
set mlpest2 nobs=nume;
if iterepocas ne 0 then do;
call symput('earlystop',left(iterepocas));
stop;
end;
else if _n_=nume and iterepocas=0 then do;iterepocas=&numeroit;
call symput('earlystop',left(iterepocas));
stop;
end;
run;

data fin;j=&earlystop;set mlpest point=j;output;stop;run;

data mlpest;set mlpest nobs=nume; if _n_=&earlystop then do;
cosa1=put(_OBJERR_,20.6) ;
cosa2=put(_VOBJERR_,20.6) ;
end;
else do;cosa1=' ';cosa2=' ';end;
run;
```

```
title1
h=2 box=1 j=c c=red 'TRAIN' c=blue '  VALIDA'
h=1.5 j=c c=black "EARLY STOPPING=&earlystop " "semilla=&semilla"
h=1 j=c c=green "NODOS OCULTOS: &ocultos  " " METODO: &meto "
"ACTIVACIÓN: &acti"
h=1 j=c c=black "EL ERROR ES EL VALOR DE LA ENTROPÍA";
;

symbol1 c=red v=circle i=join pointlabel=("#cosa1" h=1 c=red
position=bottom  j=c);
symbol2 c=blue v=circle i=join pointlabel=("#cosa2" h=1 c=blue
position=top j=c);

axis1 label=none;
proc gplot data=mlpest;plot _OBJERR_ *_iter_=1 _VOBJERR_*_iter_=2
/overlay href=&earlystop vaxis=axis1 haxis=axis1 ;run;

proc print data=fin;
var  _iter_ _OBJERR_  _AVERR_  _VNOBJ_   _VOBJ_  _VOBJERR_  _VAVERR_
;run;

%mend;


/* VALIDACION EARLY STOPPING Conjunto MINER */
%redneuronalbinaria(archivo=binario,listclass=TI_dias_afil1genero_F
  TI_tipo2  TI_estado_afiliado1  TI_edad21 TI_G_enf_totales1
  TI_OPT_edad2  TI_OPT_edad3  zona_9 zona_2 zona_4 zona_5,
listconti=edad_F edad,
vardep=costo_binario,porcen=0.50,semilla=12345,ocultos=3,meto=bprop
mom=0.2 learn=0.01,acti=TANH);

%redneuronalbinaria(archivo=binario,listclass=TI_dias_afil1genero_F
  TI_tipo2  TI_estado_afiliado1  TI_edad21 TI_G_enf_totales1
  TI_OPT_edad2  TI_OPT_edad3  zona_9 zona_2 zona_4 zona_5,
listconti=edad_F edad,
vardep=costo_binario,porcen=0.50,semilla=12345,ocultos=3,meto=levmar,a
cti=TANH);

%redneuronalbinaria(archivo=binario,listclass=TI_dias_afil1genero_F
  TI_tipo2  TI_estado_afiliado1  TI_edad21 TI_G_enf_totales1
  TI_OPT_edad2  TI_OPT_edad3  zona_9 zona_2 zona_4 zona_5,
listconti=edad_F edad,
vardep=costo_binario,porcen=0.50,semilla=12347,ocultos=3,meto=levmar,a
cti=TANH);


/* VALIDACION EARLY STOPPING Conjunto IMPORTANCIA con 3 */
%redneuronalbinaria(archivo=binario,listclass=TI_edad21
  TI_G_enf_totales1  TI_enf_totales1  TI_enf_totales2,
listconti=edadedad2  edad_M edad_F,
vardep=costo_binario,porcen=0.50,semilla=12345,ocultos=3,meto=bprop
mom=0.2 learn=0.001,acti=TANH);


%redneuronalbinaria(archivo=binario,listclass=TI_edad21
  TI_G_enf_totales1  TI_enf_totales1  TI_enf_totales2,
listconti=edadedad2  edad_M edad_F,
vardep=costo_binario,porcen=0.50,semilla=12345,ocultos=3,meto=levmar,a
cti=TANH);
```

```
%redneuronalbinaria(archivo=binario,listclass=TI_edad21
  TI_G_enf_totales1  TI_enf_totales1  TI_enf_totales2,
listconti=edadedad2  edad M edad F,
vardep=costo_binario,porcen=0.50,semilla=12347,ocultos=3,meto=levmar,a
cti=TANH);

/* VALIDACION EARLY STOPPING Conjunto IMPORTANCIA con 5 */
%redneuronalbinaria(archivo=binario,listclass=TI_edad21
  TI_G_enf_totales1  TI_enf_totales1  TI_enf_totales2,
listconti=edadedad2  edad_M edad_F,
vardep=costo_binario,porcen=0.50,semilla=12345,ocultos=5,meto=bprop
mom=0.2 learn=0.001,acti=TANH);


%redneuronalbinaria(archivo=binario,listclass=TI_edad21
  TI_G_enf_totales1  TI_enf_totales1  TI_enf_totales2,
listconti=edadedad2  edad_M edad_F,
vardep=costo_binario,porcen=0.50,semilla=12345,ocultos=5,meto=levmar,a
cti=TANH);

%redneuronalbinaria(archivo=binario,listclass=TI_edad21
  TI_G_enf_totales1  TI_enf_totales1  TI_enf_totales2,
listconti=edadedad2  edad_M edad_F,
vardep=costo_binario,porcen=0.50,semilla=12347,ocultos=5,meto=levmar,a
cti=TANH);

/* VALIDACION EARLY STOPPING Conjunto ALEATORIA 1*/
%redneuronalbinaria(archivo=binario,listclass=TI_edad21
  TI_G_enf_totales1TI_dias_afil1 TI_dias_afil2TI_estado_afiliado1
genero_F  TI_tipo2  zona_1 zona_4 zona_5 zona_9 TI_OPT_edad1
  TI_OPT_edad4,
listconti=dias_afiliacion dias_afil_porc,
vardep=costo_binario,porcen=0.50,semilla=12345,ocultos=3,meto=bprop
mom=0.2 learn=0.1,acti=TANH);

%redneuronalbinaria(archivo=binario,listclass=TI_edad21
  TI_G_enf_totales1TI_dias_afil1 TI_dias_afil2TI_estado_afiliado1
genero_F  TI_tipo2  zona_1 zona_4 zona_5 zona_9 TI_OPT_edad1
  TI_OPT_edad4,
listconti=dias_afiliacion dias_afil_porc,
vardep=costo_binario,porcen=0.50,semilla=12345,ocultos=3,meto=levmar,a
cti=TANH);

%redneuronalbinaria(archivo=binario,listclass=TI_edad21
  TI_G_enf_totales1TI_dias_afil1 TI_dias_afil2TI_estado_afiliado1
genero_F  TI_tipo2  zona_1 zona_4 zona_5 zona_9 TI_OPT_edad1
  TI_OPT_edad4,
listconti=dias_afiliacion dias_afil_porc,
vardep=costo_binario,porcen=0.50,semilla=12347,ocultos=3,meto=levmar,a
cti=TANH);


/* VALIDACION EARLY STOPPING Conjunto ALEATORIA 2 con 3 nodos*/
%redneuronalbinaria(archivo=binario,listclass=TI_edad21  TI_edad22
TI_G_enf_totales1  TI_estado_afiliado1  genero_F TI_tipo2
zona_1 zona_4 zona_5 zona_9 TI_OPT_edad1  TI_OPT_edad4,
listconti=dias_afiliacion dias_afil_porc,
vardep=costo_binario,porcen=0.50,semilla=12345,ocultos=3,meto=bprop
mom=0.2 learn=0.1,acti=TANH);
```

```
%redneuronalbinaria(archivo=binario,listclass=TI_edad21  TI_edad22
TI_G_enf_totales1  TI_estado_afiliado1  genero_F TI_tipo2
zona_1 zona_4 zona_5 zona_9 TI_OPT_edad1  TI_OPT_edad4,
listconti=dias afiliacion dias afil porc,
vardep=costo_binario,porcen=0.50,semilla=12345,ocultos=3,meto=levmar,a
cti=TANH);

%redneuronalbinaria(archivo=binario,listclass=TI_edad21  TI_edad22
TI_G_enf_totales1  TI_estado_afiliado1  genero_F TI_tipo2
zona_1 zona_4 zona_5 zona_9 TI_OPT_edad1  TI_OPT_edad4,
listconti=dias_afiliacion dias_afil_porc,
vardep=costo_binario,porcen=0.50,semilla=12347,ocultos=3,meto=levmar,a
cti=TANH);

/* VALIDACION EARLY STOPPING Conjunto ALEATORIA 2 con 7 nodos*/
%redneuronalbinaria(archivo=binario,listclass=TI_edad21  TI_edad22
TI_G_enf_totales1  TI_estado_afiliado1  genero_F TI_tipo2
zona_1 zona_4 zona_5 zona_9 TI_OPT_edad1  TI_OPT_edad4,
listconti=dias_afiliacion dias_afil_porc,
vardep=costo_binario,porcen=0.50,semilla=12345,ocultos=7,meto=bprop
mom=0.2 learn=0.1,acti=TANH);

%redneuronalbinaria(archivo=binario,listclass=TI_edad21  TI_edad22
TI_G_enf_totales1  TI_estado_afiliado1  genero_F TI_tipo2
zona_1 zona_4 zona_5 zona_9 TI_OPT_edad1  TI_OPT_edad4,
listconti=dias_afiliacion dias_afil_porc,
vardep=costo_binario,porcen=0.50,semilla=12345,ocultos=7,meto=levmar,a
cti=TANH);

%redneuronalbinaria(archivo=binario,listclass=TI_edad21  TI_edad22
TI_G_enf_totales1  TI_estado_afiliado1  genero_F TI_tipo2
zona_1 zona_4 zona_5 zona_9 TI_OPT_edad1  TI_OPT_edad4,
listconti=dias_afiliacion dias_afil_porc,
vardep=costo_binario,porcen=0.50,semilla=12347,ocultos=7,meto=levmar,a
cti=TANH);


/* VALIDACION EARLY STOPPING Conjunto mejor con 10*/
%redneuronalbinaria(archivo=binario,listclass=TI_G_enf_totales1
  TI_dias_afil1 TI_tipo2  zona_1 zona_5 zona_9 TI_OPT_edad2,
listconti=edad_F edad_M edad2,
vardep=costo_binario,porcen=0.50,semilla=12345,ocultos=3,meto=bprop
mom=0.2 learn=0.001,acti=TANH);


%redneuronalbinaria(archivo=binario,listclass=TI_G_enf_totales1
  TI_dias_afil1 TI_tipo2  zona_1 zona_5 zona_9 TI_OPT_edad2,
listconti=edad_F edad_M edad2,
vardep=costo_binario,porcen=0.50,semilla=12345,ocultos=3,meto=levmar,a
cti=TANH);

%redneuronalbinaria(archivo=binario,listclass=TI_G_enf_totales1
  TI_dias_afil1 TI_tipo2  zona_1 zona_5 zona_9 TI_OPT_edad2,
listconti=edad F edad M edad2,
vardep=costo_binario,porcen=0.50,semilla=12347,ocultos=3,meto=levmar,a
cti=TANH);

/*REDES CON VALIDACION CRUZADA */
```

```
%macro
cruzadabinarianeural(archivo=,vardepen=,conti=,categor=,ngrupos=,sinic
io=,sfinal=,nodos=,algo=,early=,
acti=,basura=c:\basura.txt,objetivo=tasafallos);
title ' ';
data final;run;
proc printto print=&basura;

/* Bucle semillas */
%do semilla=&sinicio %to &sfinal;
data dos;set &archivo;u=ranuni(&semilla);
proc sort data=dos;by u;run;
data dos (drop=nume);
retain grupo 1;
set dos nobs=nume;
if _n_>grupo*nume/&ngrupos then grupo=grupo+1;
run;
data fantasma;run;
%do exclu=1 %to &ngrupos;

data trestr tresval;
set dos;if grupo ne &exclu then output trestr;else output tresval;
PROC DMDB DATA=trestr dmdbcat=catatres;
target &vardepen;
var &conti;
class &vardepen;
%if &categor ne %then %do;class &categor &vardepen;%end;
run;
proc neural data=trestr dmdbcat=catatres random=789 ;
input &conti;
%if &categor ne %then %do;input &categor /level=nominal;%end;
target &vardepen /level=nominal;
hidden &nodos /acti=&acti;/*<<<<<******PARA DATOS LINEALES ACT=LIN
(funciónde activación lineal)
NORMALMENTE PARA DATOS NO LINEALES MEJOR ACT=TANH */
/* A PARTIR DE AQUI SON ESPECIFICACIONES DE LA RED, SE PUEDEN CAMBIAR
O AÑADIR COMO PARÁMETROS */

/*nloptions maxiter=500*/;
netoptions randist=normal ranscale=0.15 random=15459;
/* Si se desea hacer early stopping se pone prelim 0 y se marca como
comentario
la línea prelim 15...*/
/*prelim 0 */
prelim 15 preiter=10 pretech=&algo;
train maxiter=&early outest=mlpest technique=&algo;
score data=tresval role=valid out=sal ;
run;
data sal2;set sal;pro=1-%str(p_&vardepen)0;if pro>0.5 then pre11=1;
else pre11=0;run;
proc freq data=sal2;tables pre11*&vardepen/out=sal3;run;

data estadisticos (drop=count percent pre11 &vardepen);
retain vp vn fp fn suma 0;
set sal3 nobs=nume;
suma=suma+count;
if pre11=0 and &vardepen=0 then vn=count;
if pre11=0 and &vardepen=1 then fn=count;
if pre11=1 and &vardepen=0 then fp=count;
if pre11=1 and &vardepen=1 then vp=count;
if _n_=nume then do;
```

```
porcenVN=vn/suma;
porcenFN=FN/suma;
porcenVP=VP/suma;
porcenFP=FP/suma;
sensi=vp/(vp+fn);
especif=vn/(vn+fp);
tasafallos=1-(vp+vn)/suma;
tasaciertos=1-tasafallos;
precision=vp/(vp+fp);
F_M=2*Sensi*Precision/(Sensi+Precision);
output;
end;
run;
data fantasma;set fantasma estadisticos;run;
%end;
proc means data=fantasma sum noprint;var &objetivo;
output out=sumaresi sum=suma mean=media;
run;
data sumaresi;set sumaresi;semilla=&semilla;
data final (keep=suma media semilla);set final sumaresi;if suma=. then
delete;run;
%end;
proc printto ;
proc print data=final;run;
%mend;


/*REDES CONJUNTO MINER */
/*BPROP o.2 0.1 */
%cruzadabinarianeural(archivo=binario,vardepen=costo_binario,
conti=edad_F  edad,
categor=TI_dias_afil1 genero_F  TI_tipo2  TI_estado_afiliado1  TI_edad21
TI_G_enf_totales1  TI_OPT_edad2  TI_OPT_edad3  zona_9 zona_2 zona_4
  zona_5,
ngrupos=4,sinicio=12345,sfinal=12445,nodos=3,acti=tanh,early=200,algo=
bprop mom=0.2 learn=0.01);
data final8;set final;modelo=8;

%cruzadabinarianeural(archivo=binario,vardepen=costo_binario,
conti=edad_F  edad,
categor=TI_dias_afil1 genero_F  TI_tipo2  TI_estado_afiliado1  TI_edad21
TI_G_enf_totales1  TI_OPT_edad2  TI_OPT_edad3  zona_9 zona_2 zona_4
  zona_5,
ngrupos=4,sinicio=12345,sfinal=12445,nodos=3,acti=tanh,early=8,algo=le
vmar);
data final9;set final;modelo=9;

%cruzadabinarianeural(archivo=binario,vardepen=costo_binario,
conti=edad_F  edad,
categor=TI_dias_afil1 genero_F  TI_tipo2  TI_estado_afiliado1  TI_edad21
TI_G_enf_totales1  TI_OPT_edad2  TI_OPT_edad3  zona_9 zona_2 zona_4
  zona_5,
ngrupos=4,sinicio=12345,sfinal=12445,nodos=3,acti=tanh,early=200,algo=
levmar);
data final10;set final;modelo=10;

%cruzadabinarianeural(archivo=binario,vardepen=costo_binario,
conti=edad_F  edad,
categor=TI_dias_afil1 genero_F  TI_tipo2  TI_estado_afiliado1  TI_edad21
TI_G_enf_totales1  TI_OPT_edad2  TI_OPT_edad3  zona_9 zona_2 zona_4
  zona_5,
```

```
ngrupos=4,sinicio=12345,sfinal=12445,nodos=3,acti=log,early=200,algo=l
evmar);
data final11;set final;modelo=11;


%cruzadabinarianeural(archivo=binario,vardepen=costo_binario,
conti=edad_F  edad,
categor=TI_dias_afil1 genero_F  TI_tipo2  TI_estado_afiliado1  TI_edad21
TI_G_enf_totales1  TI_OPT_edad2  TI_OPT_edad3  zona_9 zona_2 zona_4
  zona_5,
ngrupos=4,sinicio=12345,sfinal=12445,nodos=3,acti=log,early=8,algo=lev
mar);
data final12;set final;modelo=12;

/*REDES CONJUNTO IMPORTANCIA*/
%cruzadabinarianeural(archivo=binario,vardepen=costo_binario,
conti=edad_F  edad,
categor=TI_dias_afil1 genero_F  TI_tipo2  TI_estado_afiliado1  TI_edad21
TI_G_enf_totales1  TI_OPT_edad2  TI_OPT_edad3  zona_9 zona_2 zona_4
  zona_5,
ngrupos=4,sinicio=12345,sfinal=12445,nodos=3,acti=tanh,early=200,algo=
bprop mom=0.2 learn=0.01);
data final13;set final;modelo=13;

%cruzadabinarianeural(archivo=binario,vardepen=costo_binario,
conti=edad_F  edad,
categor=TI_dias_afil1 genero_F  TI_tipo2  TI_estado_afiliado1  TI_edad21
TI_G_enf_totales1  TI_OPT_edad2  TI_OPT_edad3  zona_9 zona_2 zona_4
  zona_5,
ngrupos=4,sinicio=12345,sfinal=12445,nodos=5,acti=tanh,early=200,algo=
bprop mom=0.2 learn=0.01);
data final14;set final;modelo=14;

%cruzadabinarianeural(archivo=binario,vardepen=costo_binario,
conti=edad_F  edad,
categor=TI_dias_afil1 genero_F  TI_tipo2  TI_estado_afiliado1  TI_edad21
TI_G_enf_totales1  TI_OPT_edad2  TI_OPT_edad3  zona_9 zona_2 zona_4
  zona_5,
ngrupos=4,sinicio=12345,sfinal=12445,nodos=3,acti=tanh,early=200,algo=
levmar);
data final15;set final;modelo=15;

%cruzadabinarianeural(archivo=binario,vardepen=costo_binario,
conti=edad_F  edad,
categor=TI_dias_afil1 genero_F  TI_tipo2  TI_estado_afiliado1  TI_edad21
TI_G_enf_totales1  TI_OPT_edad2  TI_OPT_edad3  zona_9 zona_2 zona_4
  zona_5,
ngrupos=4,sinicio=12345,sfinal=12445,nodos=5,acti=tanh,early=200,algo=
levmar);
data final16;set final;modelo=16;

%cruzadabinarianeural(archivo=binario,vardepen=costo_binario,
conti=edad_F  edad,
categor=TI_dias_afil1 genero_F  TI_tipo2  TI_estado_afiliado1  TI_edad21
TI_G_enf_totales1  TI_OPT_edad2  TI_OPT_edad3  zona_9 zona_2 zona_4
  zona_5,
ngrupos=4,sinicio=12345,sfinal=12445,nodos=3,acti=log,early=200,algo=l
evmar);
data final17;set final;modelo=17;

%cruzadabinarianeural(archivo=binario,vardepen=costo_binario,
```

```
conti=edad_F  edad,
categor=TI_dias_afil1 genero_F  TI_tipo2  TI_estado_afiliado1   TI_edad21
TI_G_enf_totales1  TI_OPT_edad2  TI_OPT_edad3  zona_9 zona_2 zona_4
  zona 5,
ngrupos=4,sinicio=12345,sfinal=12445,nodos=5,acti=log,early=200,algo=l
evmar);
data final18;set final;modelo=18;


%cruzadabinarianeural(archivo=binario,vardepen=costo_binario,
conti=edad_F  edad,
categor=TI_dias_afil1 genero_F  TI_tipo2  TI_estado_afiliado1   TI_edad21
TI_G_enf_totales1  TI_OPT_edad2  TI_OPT_edad3  zona_9 zona_2 zona_4
  zona_5,
ngrupos=4,sinicio=12345,sfinal=12445,nodos=5,acti=log,early=10,algo=le
vmar);
data final19;set final;modelo=19;


data union;set final13 final14 final15 final16 final17 final18
final19;
proc boxplot data=union;plot media*modelo;run;

/*REDES CONJUNTO ALEATORIA 1 */

%cruzadabinarianeural(archivo=binario,vardepen=costo_binario,
conti=dias_afiliacion dias_afil_porc,
categor=TI_edad21  TI_G_enf_totales1  TI_dias_afil1 TI_dias_afil2
TI_estado_afiliado1  genero_F TI_tipo2 zona_1  zona_4 zona_5 zona_9
TI_OPT_edad1  TI_OPT_edad4,
ngrupos=4,sinicio=12345,sfinal=12445,nodos=3,acti=tanh,early=200,algo=
bprop mom=0.2 learn=0.1);
data final20;set final;modelo=20;

%cruzadabinarianeural(archivo=binario,vardepen=costo_binario,
conti=dias_afiliacion dias_afil_porc,
categor=TI_edad21  TI_G_enf_totales1  TI_dias_afil1 TI_dias_afil2
TI_estado_afiliado1  genero_F TI_tipo2 zona_1  zona_4 zona_5 zona_9
TI_OPT_edad1  TI_OPT_edad4,
ngrupos=4,sinicio=12345,sfinal=12445,nodos=3,acti=tanh,early=200,algo=
levmar);
data final21;set final;modelo=21;

%cruzadabinarianeural(archivo=binario,vardepen=costo_binario,
conti=dias_afiliacion dias_afil_porc,
categor=TI_edad21  TI_G_enf_totales1  TI_dias_afil1 TI_dias_afil2
TI_estado_afiliado1  genero_F TI_tipo2 zona_1  zona_4 zona_5 zona_9
TI_OPT_edad1  TI_OPT_edad4,
ngrupos=4,sinicio=12345,sfinal=12445,nodos=3,acti=log,early=200,algo=l
evmar);
data final22;set final;modelo=22;

data union;set final20 final21 final22;
proc boxplot data=union;plot media*modelo;run;


/*REDES CONJUNTO ALEATORIA 2 */
%cruzadabinarianeural(archivo=binario,vardepen=costo_binario,
conti=dias_afil_porc dias_afiliacion,
categor=TI_edad21 TI_edad22  TI_G_enf_totales1 TI_estado_afiliado1
genero_F  TI_tipo2 zona_1  zona_4
```

```
zona_5 zona_9 TI_OPT_edad1  TI_OPT_edad4,
ngrupos=4,sinicio=12345,sfinal=12445,nodos=3,acti=tanh,early=200,algo=
bprop mom=0.2 learn=0.1);
data final23;set final;modelo=23;

%cruzadabinarianeural(archivo=binario,vardepen=costo_binario,
conti=dias_afil_porc dias_afiliacion,
categor=TI_edad21 TI_edad22 TI_G_enf_totales1 TI_estado_afiliado1
genero_F TI_tipo2 zona_1  zona_4
zona_5 zona_9 TI_OPT_edad1  TI_OPT_edad4,
ngrupos=4,sinicio=12345,sfinal=12445,nodos=3,acti=tanh,early=11,algo=l
evmar);
data final24;set final;modelo=24;

%cruzadabinarianeural(archivo=binario,vardepen=costo_binario,
conti=dias_afil_porc dias_afiliacion,
categor=TI_edad21 TI_edad22 TI_G_enf_totales1 TI_estado_afiliado1
genero_F TI_tipo2 zona_1  zona_4
zona_5 zona_9 TI_OPT_edad1  TI_OPT_edad4,
ngrupos=4,sinicio=12345,sfinal=12445,nodos=7,acti=tanh,early=10,algo=l
evmar);
data final25;set final;modelo=25;

%cruzadabinarianeural(archivo=binario,vardepen=costo_binario,
conti=dias_afil_porc dias_afiliacion,
categor=TI_edad21 TI_edad22 TI_G_enf_totales1 TI_estado_afiliado1
genero_F TI_tipo2 zona_1  zona_4
zona_5 zona_9 TI_OPT_edad1  TI_OPT_edad4,
ngrupos=4,sinicio=12345,sfinal=12445,nodos=3,acti=log,early=11,algo=le
vmar);
data final26;set final;modelo=26;

/*REDES MEJOR CON 10 */
%cruzadabinarianeural(archivo=binario,vardepen=costo_binario,
conti=edad_F  edad_M edad2,
categor=TI_G_enf_totales1 TI_dias_afil1 TI_tipo2  zona_1 zona_5 zona_9
  TI_OPT_edad2,
ngrupos=4,sinicio=12345,sfinal=12445,nodos=3,acti=tanh,early=8,algo=bp
rop mom=0.2 learn=0.1);
data final27;set final;modelo=27;

%cruzadabinarianeural(archivo=binario,vardepen=costo_binario,
conti=edad_F  edad_M edad2,
categor=TI_G_enf_totales1 TI_dias_afil1 TI_tipo2  zona_1 zona_5 zona_9
  TI_OPT_edad2,
ngrupos=4,sinicio=12345,sfinal=12445,nodos=3,acti=tanh,early=8,algo=le
vmar);
data final28;set final;modelo=28;

%cruzadabinarianeural(archivo=binario,vardepen=costo_binario,
conti=edad_F  edad_M edad2,
categor=TI_G_enf_totales1 TI_dias_afil1 TI_tipo2  zona_1 zona_5 zona_9
  TI_OPT_edad2,
ngrupos=4,sinicio=12345,sfinal=12445,nodos=3,acti=log,early=8,algo=lev
mar);
data final29;set final;modelo=29;

data union;set final27 final28 final29;

proc boxplot data=union;plot media*modelo;run;
```

```
/* Mejor red de cada conjunto */
data union;set final12 final17 final21 final24 final29;
proc boxplot data=union;plot media*modelo;run;

/* ARBOLES - BAGGING, RANDOM SELECT Y GRADIENT BOOSTING */

/*CRUZADA RANDOM FOREST BINARIA */
%macro cruzadarandomforestbin(archivo=,vardep=,conti=,categor=,
maxtrees=100,variables=4,porcenbag=0.80,maxbranch=2,tamhoja=30,maxdept
h=10,pvalor=0.1,
ngrupos=4,sinicio=12345,sfinal=12355,objetivo=tasafallos);

data final;run;
/* Bucle semillas */
%do semilla=&sinicio %to &sfinal;

data dos;set &archivo;u=ranuni(&semilla);
proc sort data=dos;by u;run;
data dos ;
retain grupo 1;
set dos nobs=nume;
if _n_>grupo*nume/&ngrupos then grupo=grupo+1;
run;


data fantasma;run;

%do exclu=1 %to &ngrupos;
data tres;set dos;if grupo ne &exclu then vardep=&vardep;


ods listing close;
proc hpforest data=tres
maxtrees=&maxtrees
vars_to_try=&variables
trainfraction=&porcenbag
leafsize=&tamhoja
maxdepth=&maxdepth
alpha=&pvalor
exhaustive=5000
missing=useinsearch ;
target vardep/level=nominal;
input &conti/level=interval;
%if (&categor ne) %then %do;
input &categor/level=nominal;
%end;
score out=salo;
run;
ods listing ;

data salo;merge salo tres;
if p_vardep1>0.5 then pre11=1;else pre11=0;
if grupo=&exclu;
run;

proc freq data=salo;tables pre11*&vardep/out=sal3;run;
data estadisticos (drop=count percent pre11 &vardep);
retain vp vn fp fn suma 0;
set sal3 nobs=nume;
suma=suma+count;
if pre11=0 and &vardep=0 then vn=count;
```

```
if pre11=0 and &vardep=1 then fn=count;
if pre11=1 and &vardep=0 then fp=count;
if pre11=1 and &vardep=1 then vp=count;
if  n =nume then do;
porcenVN=vn/suma;
porcenFN=FN/suma;
porcenVP=VP/suma;
porcenFP=FP/suma;
sensi=vp/(vp+fn);
especif=vn/(vn+fp);
tasafallos=1-(vp+vn)/suma;
tasaciertos=1-tasafallos;
precision=vp/(vp+fp);
F_M=2*Sensi*Precision/(Sensi+Precision);
output;
end;
run;

data fantasma;set fantasma estadisticos;run;

%end;/* fin grupos */
proc means data=fantasma sum noprint;var &objetivo;
output out=sumaresi sum=suma mean=media;
run;
data sumaresi;set sumaresi;semilla=&semilla;
data final (keep=suma media semilla);set final sumaresi;if suma=. then
delete;run;
%end;/* fin semillas validación cruzada repetida*/


proc print data=final;run;


%mend;


/*CRUZADA GADIENT BOOSTING */

%macro
cruzadatreeboostbin(archivo=,vardepen=,conti=,categor=,ngrupos=,sinici
o=,sfinal=,leafsize=5,
iteraciones=100,shrink=0.01,maxbranch=2,maxdepth=4,mincatsize=15,minob
s=20,objetivo=tasafallos);
data final;run;
/* Bucle semillas */
%do semilla=&sinicio %to &sfinal;
data dos;set &archivo;u=ranuni(&semilla);
proc sort data=dos;by u;run;
data dos ;
retain grupo 1;
set dos nobs=nume;
if _n_>grupo*nume/&ngrupos then grupo=grupo+1;
run;
data fantasma;run;
%do exclu=1 %to &ngrupos;
data tres;set dos;if grupo ne &exclu then vardep=&vardepen;

proc treeboost data=tres
exhaustive=1000 intervaldecimals=max
leafsize=&leafsize iterations=&iteraciones maxbranch=&maxbranch
```

```
maxdepth=&maxdepth mincatsize=&mincatsize missing=useinsearch
shrinkage=&shrink
splitsize=&minobs;
%if (&categor ne) %then %do;
input &categor/level=nominal;
%end;
input &conti/level=interval;
target vardep /level=binary;
save fit=iteraciones importance=impor model=modelo rules=reglas;
subseries largest;
score out=sal;

data sal2;set sal;pro=1-p_vardep0;if pro>0.5 then pre11=1; else
pre11=0;
if grupo=&exclu then output;run;
proc freq data=sal2;tables pre11*&vardepen/out=sal3;run;
data estadisticos (drop=count percent pre11 &vardepen);
retain vp vn fp fn suma 0;
set sal3 nobs=nume;
suma=suma+count;
if pre11=0 and &vardepen=0 then vn=count;
if pre11=0 and &vardepen=1 then fn=count;
if pre11=1 and &vardepen=0 then fp=count;
if pre11=1 and &vardepen=1 then vp=count;
if _n_=nume then do;
porcenVN=vn/suma;
porcenFN=FN/suma;
porcenVP=VP/suma;
porcenFP=FP/suma;
sensi=vp/(vp+fn);
especif=vn/(vn+fp);
tasafallos=1-(vp+vn)/suma;
tasaciertos=1-tasafallos;
precision=vp/(vp+fp);
F_M=2*Sensi*Precision/(Sensi+Precision);
output;
end;
run;

data fantasma;set fantasma estadisticos;run;
%end;
proc means data=fantasma sum noprint;var &objetivo;
output out=sumaresi sum=suma mean=media;
run;
data sumaresi;set sumaresi;semilla=&semilla;
data final (keep=suma media semilla);set final sumaresi;if suma=. then
delete;run;
%end;
proc print data=final;run;
%mend;


/*BAGGING ALEATORIA 2*/
%cruzadarandomforestbin(archivo=binario,vardep=costo_binario,conti=dia
s_afil_porc dias_afiliacion,
categor=TI_edad21  TI_edad22 TI_G_enf_totales1  TI_estado_afiliado1
  genero_F  TI_tipo2  zona_1  zona_4  zona_5  zona_9 TI_OPT_edad1
  TI_OPT_edad4,
maxtrees=1000,variables=14,porcenbag=0.40,maxbranch=2,tamhoja=30,maxde
pth=10,pvalor=0.1,
ngrupos=4,sinicio=13345,sfinal=13395,objetivo=tasafallos);
```

```
data final30;set final;modelo=30;


/*BAGGING porcentbag */
%cruzadarandomforestbin(archivo=binario,vardep=costo_binario,conti=dia
s_afil_porc dias_afiliacion,
categor=TI_edad21  TI_edad22 TI_G_enf_totales1  TI_estado_afiliado1
  genero_F  TI_tipo2  zona_1  zona_4  zona_5  zona_9 TI_OPT_edad1
  TI_OPT_edad4,
maxtrees=1000,variables=14,porcenbag=0.70,maxbranch=2,tamhoja=30,maxde
pth=10,pvalor=0.1,
ngrupos=4,sinicio=13345,sfinal=13395,objetivo=tasafallos);
data final31;set final;modelo=31;

%cruzadarandomforestbin(archivo=binario,vardep=costo_binario,conti=dia
s_afil_porc dias_afiliacion,
categor=TI_edad21  TI_edad22 TI_G_enf_totales1  TI_estado_afiliado1
  genero_F  TI_tipo2  zona_1  zona_4  zona_5  zona_9 TI_OPT_edad1
  TI_OPT_edad4,
maxtrees=1000,variables=14,porcenbag=0.70,maxbranch=2,tamhoja=30,maxde
pth=20,pvalor=0.1,
ngrupos=4,sinicio=13345,sfinal=13395,objetivo=tasafallos);
data final32;set final;modelo=32;


%cruzadarandomforestbin(archivo=binario,vardep=costo_binario,conti=dia
s_afil_porc dias_afiliacion,
categor=TI_edad21  TI_edad22 TI_G_enf_totales1  TI_estado_afiliado1
  genero_F  TI_tipo2  zona_1  zona_4  zona_5  zona_9 TI_OPT_edad1
  TI_OPT_edad4,
maxtrees=1000,variables=14,porcenbag=0.70,maxbranch=2,tamhoja=30,maxde
pth=20,pvalor=0.01,
ngrupos=4,sinicio=13345,sfinal=13395,objetivo=tasafallos);
data final33;set final;modelo=33;

%cruzadarandomforestbin(archivo=binario,vardep=costo_binario,conti=dia
s_afil_porc dias_afiliacion,
categor=TI_edad21  TI_edad22 TI_G_enf_totales1  TI_estado_afiliado1
  genero_F  TI_tipo2  zona_1  zona_4  zona_5  zona_9 TI_OPT_edad1
  TI_OPT_edad4,
maxtrees=1000,variables=14,porcenbag=0.70,maxbranch=2,tamhoja=30,maxde
pth=10,pvalor=0.3,
ngrupos=4,sinicio=13345,sfinal=13395,objetivo=tasafallos);
data final34;set final;modelo=34;

data union;set final30 final31 final32 final33 final34;
proc boxplot data=union;plot media*modelo;run;
```

Modelización variable objetivo continua coste total

```
libname Discoc 'C:\';
data uno;set discoc.costomedio;run;

data uno;set discoc.costetotal;run;
%macro
randomselect(data=,listclass=,vardepen=,modelo=,criterio=,sinicio=,sfi
nal=,fracciontrain=,directorio=&directorio);
options nocenter linesize=256;
proc printto print="&directorio\kk.txt";run;
data _null_;file "&directorio\cosa.txt" linesize=2000;run;
%do semilla=&sinicio %to &sfinal;
```

```
proc surveyselect data=&data rate=&fracciontrain out=sal1234
seed=&semilla;run;
ods output   SelectionSummary=modelos;
ods output    SelectedEffects=efectos;
ods output    Glmselect.SelectedModel.FitStatistics=ajuste;
proc glmselect data=sal1234  plots=all seed=&semilla;
class &listclass;
model &vardepen= &modelo/ selection=stepwise(select=&criterio
choose=&criterio) details=all stats=all;
run;
ods graphics off;
ods html close;
data union;i=5;set efectos;set ajuste point=i;run;
data _null_;semilla=&semilla;file "&directorio\cosa.txt" mod
linesize=2000;set union;put effects ;run;
%end;
proc printto ;run;
data todos;
infile "&directorio\cosa.txt" linesize=2000;
length efecto $ 1000;
input efecto @@;
if efecto ne 'Intercept' then output;
run;
proc freq data=todos;tables efecto /out=sal;run;
proc sort data=sal;by descending count;
proc print data=sal;run;

data todos;
infile "&directorio\cosa.txt" linesize=2000;
length efecto $ 1000;
input efecto $ &&;
run;
proc freq data=todos;tables efecto /out=salefec;run;
proc sort data=salefec;by descending count;
proc print data=salefec;run;
data _null_;set salefec;put efecto;run;
%mend;


%randomselect(data=uno,directorio=Discoc,
listclass=TI_edad21 TI_edad22 TI_G_enf_totales1 TI_dias_afil1
TI_dias_afil2 TI_OPT_edad21 TI_OPT_edad22 TI_OPT_edad24
TI_enf_totales1 TI_enf_totales2 TI_estado_afiliado1 genero_F TI_tipo1
TI_tipo2 VIH dialisis menos1 oncologia_adultos reumatologia_colageno
zona_1 zona_10 zona_11 zona_2 zona_4 zona_5 zona_9 TI_OPT_edad1
  TI_OPT_edad2  TI_OPT_edad3  TI_OPT_edad4,
vardepen=costo_medio,
modelo=edad_F edad_M dias_afiliacion dias_afil_porc edad edad2
TI_edad21 TI_edad22 TI_G_enf_totales1 TI_dias_afil1 TI_dias_afil2
TI_OPT_edad21 TI_OPT_edad22 TI_OPT_edad24 TI_enf_totales1
TI_enf_totales2 TI_estado_afiliado1 genero_F TI_tipo1 TI_tipo2 VIH
dialisis menos1 oncologia_adultos reumatologia_colageno zona_1 zona_10
zona_11 zona_2 zona_4 zona_5 zona_9 TI_OPT_edad1  TI_OPT_edad2
  TI OPT edad3  TI_OPT_edad4,
criterio=AIC,
sinicio=12345,
sfinal=12545,
fracciontrain=0.8);
```

```
/* Se realiza random select con las variables pre seleccionadas en el
apunto anterior y se analiza cuantas veces son seleccionados el set de
datos */

%randomselect(data=uno,directorio=Discoc,
listclass=TI_edad21 TI_edad22 TI_G_enf_totales1 TI_dias_afil1
TI_dias_afil2 TI_OPT_edad21 TI_OPT_edad22 TI_OPT_edad24
TI_enf_totales1 TI_enf_totales2 TI_estado_afiliado1 genero_F TI_tipo1
TI_tipo2 VIH dialisis menos1 oncologia_adultos reumatologia_colageno
zona_1 zona_10 zona_11 zona_2 zona_4 zona_5 zona_9 TI_OPT_edad1
  TI_OPT_edad2  TI_OPT_edad3  TI_OPT_edad4,
vardepen=costo_medio,
modelo=edad_F edad_M dias_afiliacion dias_afil_porc edad edad2
TI_edad21 TI_edad22 TI_G_enf_totales1 TI_dias_afil1 TI_dias_afil2
TI_OPT_edad21 TI_OPT_edad22 TI_OPT_edad24 TI_enf_totales1
TI_enf_totales2 TI_estado_afiliado1 genero_F TI_tipo1 TI_tipo2 VIH
dialisis menos1 oncologia_adultos reumatologia_colageno zona_1 zona_10
zona_11 zona_2 zona_4 zona_5 zona_9 TI_OPT_edad1  TI_OPT_edad2
  TI_OPT_edad3  TI_OPT_edad4,
criterio=BIC,
sinicio=12345,
sfinal=12445,
fraccion train=0.8);
/*

Los mejores 5 set de datos se corren con validacion cruzada repetida
para seleccionar el mejor modelo
*/

/*

%macro
cruzada(archivo=,vardepen=,conti=,categor=,ngrupos=,sinicio=,sfinal=);
data final;run;

%do semilla=&sinicio %to &sfinal;
data dos;set &archivo;u=ranuni(&semilla);
proc sort data=dos;by u;run;
data dos;
retain grupo 1;
set dos nobs=nume;
if _n_>grupo*nume/&ngrupos then grupo=grupo+1;
run;
data fantasma;run;
%do exclu=1 %to &ngrupos;
data tres;set dos;if grupo ne &exclu then vardep=&vardepen;
proc glm data=tres noprint;
%if &categor ne %then %do;class &categor;model vardep=&conti
&categor;%end;
%else %do;model vardep=&conti;%end;
output out=sal p=predi;run;
data sal;set sal;resi2=(&vardepen-predi)**2;if grupo=&exclu then
output;run;
data fantasma;set fantasma sal;run;
%end;
proc means data=fantasma sum noprint;var resi2;
output out=sumaresi sum=suma mean=media;
run;
data sumaresi;set sumaresi;semilla=&semilla;
data final (keep=suma media semilla);set final sumaresi;if suma=. then
delete;run;
```

```
%end;
proc print data=final;run;
%mend;
*/

/* Modelo 1 seleccion miner */
%cruzada(archivo=uno,vardepen=costo_medio,
conti=edad2 edad edad_F ,
categor=oncologia_adultos TI_G_enf_totales1  TI_OPT_edad22
  TI_OPT_edad4  TI_tipo2  zona_2 dialisis,
ngrupos=10,sinicio=12345,sfinal=12545);
data final1;set final;modelo=1;

/*Modelo 2 importancia*/
%cruzada(archivo=uno,vardepen=costo_medio,
conti=edad  edad2  edad_M ,
categor=dialisis TI_G_enf_totales1  TI_enf_totales1  TI_enf_totales2
  VIH  reumatologia_colageno,
ngrupos=10,sinicio=12345,sfinal=12545);
data final2;set final;modelo=2;

/*Modelo 3 random 1*/
%cruzada(archivo=uno,vardepen=costo_medio,
conti=dias_afiliacion edad2,
categor=menos1 dialisis  oncologia_adultos  reumatologia_colageno VIH
  TI_enf_totales1  TI_enf_totales2  TI_OPT_edad3,
ngrupos=10,sinicio=12345,sfinal=12445);
data final3;set final;modelo=3;

/*Modelo 4*random 2 */
%cruzada(archivo=uno,vardepen=costo_medio,
conti=dias_afiliacion edad2,
categor=menos1 dialisis  oncologia_adultos  reumatologia_colageno
  TI_enf_totales1  TI_enf_totales2  TI_OPT_edad4,
ngrupos=10,sinicio=12345,sfinal=12445);
data final4;set final;modelo=4;

data union;set final1 final2 final3 final4;
proc boxplot data=union;plot media*modelo;run;


/* REDES */

/*****************************************************************
****
/* SI SE QUIERE COMPARAR POR EJEMPLO NÚMERO DE NODOS POR VALIDACIÓN
CRUZADA Y BOXPLOT
/*****************************************************************
***/

%macro
cruzadaneural(archivo=,vardepen=,conti=,categor=,ngrupos=,sinicio=,sfi
nal=,ocultos=3,algo=levmar,acti=tanh,early=,directorio=);
/*Si no se quiere información en output usar esto (cambiar el archivo
de destino):
proc printto print='&directorio\basura.txt';
C:\Users\felip\Desktop\TFM R
*/
proc printto print="&directorio\basura.txt";
data final;run;
%do semilla=&sinicio %to &sfinal;
```

```
data dos;set &archivo;u=ranuni(&semilla);
proc sort data=dos;by u;run;
data dos (drop=nume);
retain grupo 1;
set dos nobs=nume;
if _n_>grupo*nume/&ngrupos then grupo=grupo+1;
run;
data fantasma;run;
%do exclu=1 %to &ngrupos;
data trestr tresval;
set dos;if grupo ne &exclu then output trestr;else output tresval;
PROC DMDB DATA=trestr dmdbcat=catatres;
target &vardepen;
var &vardepen &conti;
%if &categor ne %then %do;class &categor;%end;
run;
proc neural data=trestr dmdbcat=catatres random=789 ;
input &conti;
%if &categor ne %then %do;input &categor /level=nominal;%end;
target &vardepen;
hidden &ocultos /act=&acti;/*<<<<<******PARA DATOS LINEALES ACT=LIN
(funciónde activación lineal)
NORMALMENTE PARA DATOS NO LINEALES MEJOR ACT=TANH */
/* A PARTIR DE AQUI SON ESPECIFICACIONES DE LA RED, SE PUEDEN CAMBIAR
O AÑADIR COMO PARÁMETROS */

/* ESTO ES PARA EARLY STOPPING (maxiter=numero de iteraciones
limitado)*/

%if &early ne %then %do;
nloptions maxiter=&early;
netoptions randist=normal ranscale=0.1 random=15115;%end;
/* %else %do;prelim 10;%end;*/
%if &early ne %then %do;
train maxiter=&early /* early stopping cambiar maxiter=25 por ejemplo
*/ outest=mlpest technique=&algo;%end;
%else %do;train maxiter=100 /* early stopping cambiar maxiter=25 por
ejemplo */ outest=mlpest technique=&algo/* bprop mom=0.2
learn=0.1*/;%end;
score data=tresval role=valid out=sal ;
run;
data sal;set sal;resi2=(p_&vardepen-&vardepen)**2;run;
data fantasma;set fantasma sal;run;
%end;
proc means data=fantasma sum noprint;var resi2;
output out=sumaresi sum=suma mean=media;
run;
data sumaresi;set sumaresi;semilla=&semilla;
data final (keep=suma media semilla);set final sumaresi;if suma=. then
delete;run;
%end;
proc printto;run;
proc print data=final;run;
%mend;

%macro
nodosvalcruza(ini=,fin=,increme=,data=,vardepen=,conti=,categor=,acti=
);
%do nod=&ini %to &fin %by &increme;
%cruzadaneural(archivo=&data,vardepen=&vardepen,conti=&conti,categor=&
categor,
```

```
acti=&acti,ngrupos=4,sinicio=12345,sfinal=12395,ocultos=&nod,algo=levm
ar);
data finaln&nod;set final;modelo=&nod;run;
%end;
data union;set %do i=&ini %to &fin %by &increme; finaln&i %end;;;;
%mend;

/* para comprobar numero de nodos optimo a utilizar esta con Levmar
por defecto*/
/*MINER */
%nodosvalcruza(ini=3,fin=15,increme=3,data=uno,vardepen=costo_medio,co
nti=edad2 edad edad_F,
categor=oncologia_adultos TI_G_enf_totales1  TI_OPT_edad22
  TI_OPT_edad4  TI_tipo2  zona_2 dialisis,acti=tanh);
proc boxplot data=Union;plot media*modelo;
run;

/*IMPORTANCIA */
%nodosvalcruza(ini=3,fin=18,increme=3,data=uno,vardepen=costo_medio,co
nti=edad  edad2,
categor=TI_G_enf_totales1 TI_enf_totales1  TI_enf_totales2  VIH
  dialisis  oncologia_adultos,acti=tanh);
proc boxplot data=Union;plot media*modelo;
run;

/*ALEATORIA 1*/
%nodosvalcruza(ini=3,fin=15,increme=3,data=uno,vardepen=costo_medio,co
nti=dias_afiliacion  edad2,
categor=menos1dialisis  oncologia_adultos  reumatologia_colagenoVIH
  TI_enf_totales1  TI_enf_totales2  TI_OPT_edad3,acti=tanh);
proc boxplot data=Union;plot media*modelo;
run;


/*ALEATORIA 2*/
%nodosvalcruza(ini=3,fin=15,increme=3,data=uno,vardepen=costo_medio,co
nti=dias_afiliacion  edad2,
categor=menos1dialisis  oncologia_adultos  reumatologia_colageno
  TI_enf_totales1  TI_enf_totales2  TI_OPT_edad4,acti=tanh);
proc boxplot data=Union;plot media*modelo;
run;


/*para probar con BPROP correr este codigo
%macro
nodosvalcruza(ini=,fin=,increme=,data=,vardepen=,conti=,categor=,acti=
);
%do nod=&ini %to &fin %by &increme;
%cruzadaneural(archivo=&data,vardepen=&vardepen,conti=&conti,categor=&
categor,
acti=&acti,ngrupos=4,sinicio=12345,sfinal=12395,ocultos=&nod,algo=bpro
p);
data finaln&nod;set final;modelo=&nod;run;
%end;
data union;set %do i=&ini %to &fin %by &increme; finaln&i %end;;;;
%mend;


/*EARLY STOPPING CON NODOS DE R */
```

```
%macro
redneuronal(archivo=,listclass=,listconti=,vardep=,porcen=,semilla=,oc
ultos=,algo=,acti=);

%if &listclass eq %then %do;

PROC DMDB DATA=&archivo dmdbcat=catauno;
target &vardep;
var &listconti &vardep;
run;
%end;
%else %do;
PROC DMDB DATA=&archivo dmdbcat=catauno;
target &vardep;
var &listconti &vardep;
class &listclass;
run;
%end;

data ooo;set &archivo;run;
data datos;set ooo nobs=nume;tr=int(&porcen*nume);call
symput('tr',left(tr));u=ranuni(&semilla);run;
proc sort data=datos;by u;run;
data datos valida;set datos;if _n_>tr then output valida;else output
datos;run;

proc neural data=datos dmdbcat=catauno validata=valida graph;
input &listconti / id=i;
input &listclass / level=nominal;
target &vardep / id=o;
hidden &ocultos / id=h act=&acti;
nloptions maxiter=10000;
netoptions randist=normal ranscale=0.1 random=15115;
train maxiter=10000 outest=mlpest estiter=1 technique=&algo;
score data=datos out=mlpout outfit=mlpfit;
score data=valida out=mlpout2 outfit=mlpfit2 role=valid;
run;

data mlpest2 ;
k=3;
retain iterepocas 0;
set mlpest;
eval=_VOBJERR_;
x3=lag3(eval);
x6=lag6(eval);
if _n_>6 and eval>x3 and eval>x6 then iterepocas=_n_;
run;

data;
set mlpest2 nobs=nume;
if iterepocas ne 0 then do;control=1;
call symput('earlystop',left(iterepocas));
stop;
end;
if _n_=nume and control ne 1 then do;
ka=0;
call symput('earlystop',left(ka));
end;
run;

data fin;j=&earlystop;set mlpest point=j;output;stop;run;
```

```
data mlpest;set mlpest nobs=nume; if _n_=&earlystop then do;
cosa1=put(_OBJERR_,20.6) ;
cosa2=put(_VOBJERR_,20.6) ;
end;
else do;cosa1=' ';cosa2=' ';end;
run;

title1
h=2 box=1 j=c c=red 'TRAIN' c=blue '  VALIDA'
h=1.5 j=c c=black "EARLY STOPPING=&earlystop " "semilla=&semilla"
h=1 j=c c=green "NODOS OCULTOS: &ocultos  " " METODO: &algo "
"ACTIVACIÓN: &acti";
;

symbol1 c=red v=circle i=join pointlabel=("#cosa1" h=1 c=red
position=bottom  j=c);
symbol2 c=blue v=circle i=join pointlabel=("#cosa2" h=1 c=blue
position=top j=c);

axis1 label=none;
proc gplot data=mlpest;plot _OBJERR_ *_iter_=1 _VOBJERR_*_iter_=2
/overlay href=&earlystop vaxis=axis1 haxis=axis1 ;run;

proc print data=fin;
var  _iter_ _OBJERR_  _AVERR_  _VNOBJ_   _VOBJ_  _VOBJERR_  _VAVERR_
;run;

%mend;

/* EXPLORAR EL EARLY STOPPING MODELO 4-  esto es para probar early
stopping con bprop y/o levmar*/
/*MINER */
%redneuronal(archivo=uno,listclass=oncologia_adultosTI_G_enf_totales1
   TI_OPT_edad22 TI_OPT_edad4  TI_tipo2  zona_2
   dialisis,listconti=edad2  edad edad_F,
vardep=costo_medio,porcen=0.80,semilla=12345,ocultos=15,algo=BPROP
MOM=0.2 LEARN=0.1,acti=TANH);

%redneuronal(archivo=uno,listclass=oncologia_adultosTI_G_enf_totales1
   TI_OPT_edad22 TI_OPT_edad4  TI_tipo2  zona_2
   dialisis,listconti=edad2  edad edad_F,
vardep=costo_medio,porcen=0.80,semilla=12345,ocultos=15,algo=levmar,ac
ti=TANH);


/*IMPORTANCIA */
%nodosvalcruza(ini=3,fin=18,increme=3,data=uno,vardepen=costo_medio,co
nti=edad  edad2,
categor=TI_G_enf_totales1 TI_enf_totales1  TI_enf_totales2  VIH
   dialisis  oncologia_adultos,acti=tanh);
proc boxplot data=Union;plot media*modelo;
run;

/*ALEATORIA 1*/
%nodosvalcruza(ini=3,fin=15,increme=3,data=uno,vardepen=costo_medio,co
nti=dias_afiliacion  edad2,
categor=menos1dialisis  oncologia_adultos  reumatologia_colagenoVIH
   TI_enf_totales1  TI_enf_totales2  TI_OPT_edad3,acti=tanh);
proc boxplot data=Union;plot media*modelo;
run;
```

```
/*ALEATORIA 2*/
%nodosvalcruza(ini=3,fin=15,increme=3,data=uno,vardepen=costo_medio,co
nti=dias afiliacion  edad2,
categor=menos1dialisis  oncologia_adultos  reumatologia_colageno
  TI_enf_totales1  TI_enf_totales2  TI_OPT_edad4,acti=tanh);
proc boxplot data=Union;plot media*modelo;
run;

/*IMPORTANCIA */
%cruzadaneural(archivo=uno,vardepen=costo_medio,
conti=edad  edad2 edad_F,
categor=dialisis oncologia_adultos  TI_enf_totales2  TI_G_enf_totales1
VIH genero_F  reumatologia_colageno,
ngrupos=10,sinicio=12345,sfinal=12395,ocultos=20,algo=levmar,acti=tanh
,early=,directorio=);
data final6;set final;modelo=6;

%cruzadaneural(archivo=uno,vardepen=costo_medio,
conti=edad  edad2 edad_F,
categor=dialisis oncologia_adultos  TI_enf_totales2  TI_G_enf_totales1
VIH genero_F  reumatologia_colageno,
ngrupos=10,sinicio=12345,sfinal=12395,ocultos=20,algo=bprop mom=0.2
learn=0.1,acti=tanh,early=,directorio=);
data final7;set final;modelo=7;

%cruzadaneural(archivo=uno,vardepen=costo_medio,
conti=edad  edad2 edad_F,
categor=dialisis oncologia_adultos  TI_enf_totales2  TI_G_enf_totales1
VIH genero_F  reumatologia_colageno,
ngrupos=10,sinicio=12345,sfinal=12395,ocultos=20,algo=levmar,acti=tanh
,early=20,directorio=);
data final8;set final;modelo=8;

%cruzadaneural(archivo=uno,vardepen=costo_medio,
conti=edad  edad2 edad_F,
categor=dialisis oncologia_adultos  TI_enf_totales2  TI_G_enf_totales1
VIH genero_F  reumatologia_colageno,
ngrupos=10,sinicio=12345,sfinal=12395,ocultos=20,algo=bprop mom=0.2
learn=0.1,acti=tanh,early=7,directorio=);
data final9;set final;modelo=9;

%cruzadaneural(archivo=uno,vardepen=costo_medio,
conti=edad  edad2 edad_F,
categor=dialisis oncologia_adultos  TI_enf_totales2  TI_G_enf_totales1
VIH genero_F  reumatologia_colageno,
ngrupos=10,sinicio=12345,sfinal=12395,ocultos=10,algo=levmar,acti=tanh
,early=,directorio=);
data final10;set final;modelo=10;

%cruzadaneural(archivo=uno,vardepen=costo_medio,
conti=edad  edad2 edad_F,
categor=dialisis oncologia_adultos  TI_enf_totales2  TI_G_enf_totales1
VIH genero F  reumatologia colageno,
ngrupos=10,sinicio=12345,sfinal=12395,ocultos=20,algo=levmar,acti=log,
early=,directorio=);
data final11;set final;modelo=11;

%cruzadaneural(archivo=uno,vardepen=costo_medio,
conti=edad  edad2 edad_F,
```

```
categor=dialisis oncologia_adultos  TI_enf_totales2  TI_G_enf_totales1
VIH genero_F  reumatologia_colageno,
ngrupos=10,sinicio=12345,sfinal=12395,ocultos=15,algo=levmar,acti=tanh
,early=,directorio=);
data final12;set final;modelo=12;

/*importancia */
%cruzadaneural(archivo=uno,vardepen=costo_medio,
conti=edad  edad2 edad_M,
categor=dialisis TI_G_enf_totales1 TI_enf_totales1 TI_enf_totales2
  VIH reumatologia_colageno,
ngrupos=10,sinicio=12345,sfinal=12395,ocultos=20,algo=levmar,acti=tanh
,early=,directorio=);
data final13;set final;modelo=13;

/*random 1*/
%cruzadaneural(archivo=uno,vardepen=costo_medio,
conti=edad  edad2  edad_F,
categor=reumatologia_colageno  VIH  dialisis  oncologia_adultos
  TI_dias_afil2 TI_OPT_edad21 TI_OPT_edad22 TI_enf_totales1
  TI_enf_totales2  genero_F,
ngrupos=10,sinicio=12345,sfinal=12395,ocultos=20,algo=levmar,acti=tanh
,early=,directorio=);
data final14;set final;modelo=14;

/*random 2*/
%cruzadaneural(archivo=uno,vardepen=costo_medio,
conti=edad  edad2  edad_F,
categor=reumatologia_colageno  VIH  dialisis  oncologia_adultos
  TI_dias_afil2 TI_OPT_edad21 TI_OPT_edad22 TI_enf_totales1
  TI_enf_totales2  genero_F  TI_tipo1,
ngrupos=10,sinicio=12345,sfinal=12395,ocultos=20,algo=levmar,acti=tanh
,early=,directorio=);
data final15;set final;modelo=15;

data union;set final6 final13 final14 final15;
proc boxplot data=union;plot media*modelo;
run;

/* final7 final8 final9 final10 final11 final12 final13 final4 final6
final14 final15 final17 final18 final19*/
```